Jurassic News

Retrocomputer Magazine

Anno 3 - Numero 16 - Aprile 2008

ALTAIR 8800

Prima dei primi home

BASCOM per Apple

Il calcolatore ibrido

ABAP

Olivetti M10



Jurassic News

Rivista aperiodica di Retro-computing

Coordinatore editoriale
Salvatore Macomer [Sm]

Redazione
Sonicher [Sn]
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Tullio Nicolussi [Tn]
Lorenzo 2 [L2]
Besdelsec [Bs]
Maurizio Martone [Mm]
Mister X [Mx]
Alberta [Alb]
Gianni [Mg]

Impaginazione e grafiea
Anna [An]

Diffusione
marketing@jurassicnews.com

La rivista viene diffusa in formato PDF via Internet agli utenti registrati sul sito www.jurassicnews.com
la registrazione è gratuita e anonima; si gradisce comunque una registrazione nominativa.

Contatti
info@jurassicnews.com

Jurassic News promuove la libera circolazione delle idee

Aprile 2008

In Copertina

Un'altra macchina dell'italiana Olivetti è protagonista della nostra recensione hardware. L'M10 è a buon titolo considerato uno dei primi veri portatili che hanno calcato la scena dell'informatica al mondo (scusate se è poco!).

Editoriale

Numero ricco, mi ci ficco...

Questo fascicolo di JN sono sicuro vi piacerà! Le pagine magari sono un po' meno rispetto ai numeri passati, ma gli argomenti sono tanti e tutti (speriamo) interessanti.

Sappiamo che è difficile soddisfare tutti i palati e non solo per noi: in ogni aspetto dello scibile umano si può dire che esistano (e per fortuna che ci sono) due fazioni: i favorevoli e i contrari, quelli che lodano e quelli che imbrodano, insomma gli amici e i nemici.
Tutta questa "tirata" per invitarvi ad intervenire sulla rivista: come la vorreste, cosa vi piacerebbe trovarci, possibilmente con critiche costruttive...
Fate uno sforzo, sù!

Su questo numero troverete la seconda parte dell'articolo "Prima dei primi home" e la recensione dell'M10 della Olivetti. Per l'emulazione una entry pesante: nientemeno che l'ALTAIR 8800, una delle macchine simbolo.
Il Club Apple continua con il compilatore BASCOM, in fondo una naturale continuazione dell'articolo precedente che trattava del BASIC Microsoft.
Il settore "teorico" della rivista rispolvera gli algoritmi di sort, nell'intento di farne una trattazione la più completa possibile.
Ma è della rubrica "Laboratorio" che voglio raccomandarvi l'articolo: avete mai sentito parlare dei computer ibridi? Ecco come il nostro collaboratore Gianni vi spiegherà di cosa si tratta.

Buona lettura a tutti,

[Sm]

Jurassic News

è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Retrocomputing

La sòla

Frequentando il mercato dell'usato si incappa prima o poi nella classica fregatura. Accettarla o ribellarsi?

Fratello Imbroglio, sorella Sòla... parafrasando il ben più nobile scritto di San Francesco d'Assisi, tale "Cantico delle Creature" che assolve con buona grazia anche i meno desiderabili compagni della nostra avventura nel mondo terreno.

E' nell'ordine delle cose che chi fruga fra la merce rimasta in magazzino, se non addirittura già avviata alla discarica, possa incappare nell'oggetto guasto e per di più irreparabile. Questo ci stà, anzi fa parte del gioco e viene accettato di buon grado. Si potrebbe addirittura affermare che le delusioni accrescono la soddisfazione del recupero che, magari una volta su dieci, si conclude felicemente.

Logica vorrebbe che gli oggetti che vengono buttati siano inutilizzabili, ma questo è vero solo in parte nella nostra società basata sul consumo. In realtà la gente si libera delle cose non tanto perché esse sono giunte alla fine della loro vita di servizio, quanto perché la società impone che siano rimpiazzate con qualcosa di più moderno.

Intendiamoci, un certo livello di evoluzione è utile e contribuisce anche a migliorare la nostra qualità della vita. Quello che non è sano è il cambio delle cose non per bisogno pratico ma per bisogno psicologico. Cambiamo macchina perché gli altri hanno il SUV, prendiamo il televisore al plasma perché ci sembra che il salotto ne guadagni in prestigio e via elencando.

Poi c'é il problema degli spazi. Non per nulla molti home si recuperano a valle di traslochi di amici e conoscenti o a fronte di classiche pulizie dello scantinato intraprese dal vicino di casa che magari deve farci entrare il gommone.

Quante volte abbiamo fatto dei semplici passamano con alcuni oggetti che ci sembravano utili e poi abbiamo dovuto a nostre spese riciclare? Oppure quanti viaggi in discarica abbiamo fatto perché il furbo nostro interlocutore ci ha mollato l'Atari ST solo se gli portavamo via anche il televisore bianco-nero e la macchina da cucire della suocera defunta?

Tutto questo, lo ripetiamo, fa parte del gioco e perciò ben accolto. Chi si lamenta del tempo perduto per portare a casa un vecchio home regalato che poi si è rivelato guasto, non ha capito nulla di retro computing!

Diverso, e veniamo a dire il merito, il caso in cui la macchina non funzionante l'abbiamo comprata o, caso ancora più grave dal mio punto di vista, scambiata con analogo oggetto funzionante.

L'acquisto, soprattutto se fatto nei mercatini dove i privati espongono le loro cose tirate fuori dalla soffitta,

è fonte di frustrazione. Infatti il venditore sicuramente ci avrà spacciato l'oggetto come funzionante al 100%, magari in buona fede (ma ne dubitiamo). Scoprire poi a casa che quello che sembrava un affare, venti Euro per una Amiga 600 con una decina di giochi su floppy, poi si rivela la classica sòla, è frustrante. E chi lo recupera più il venditore? Sì, magari il mese prossimo al mercatino c'è ancora, ma fingerà di non conoscervi, giurerà che computer lui non ne ha mai venduto e se proprio costretto vi ribalterà l'onere della prova affermando che quando lui l'ha provato la macchina funzionava e che "... sicuramente si è guastata durante il trasporto quando l'avete portata a casa...".

In qualche caso ci si può consolare con la considerazione che magari venti Euro sono tanti per dieci giochini ma che in fondo l'alimentatore funziona e qualche tasto di riserva è sempre utile... Rimane l'amaro in bocca non tanto per la disillusione che ci conferma di vivere in un mondo popolato da "furbi" e maggiormente per la delusione di non poter raccontare del recupero e di non poter annoverare come funzionante quel pezzo che mancava nella propria collezione.

La cosa si fa grave quanto la fregatura, chiamiamola con il suo vero nome perbacco, ce la molla un "amico" o collega retro computerista. Allora sì che è dura da mandare giù! Ma come: condividiamo la stessa passione, apparteniamo antropologicamente alla stessa tribù culturale e hai il coraggio di pugnalarmi alle spalle?

Personalmente diffido da chiunque non si esponga spontaneamente per eventuali rivalse, colui cioè che non ti assicura un recapito o che, almeno a parole di rassicura sulla possibilità di riavere indietro i nostri soldi se la macchina non funziona. Questo è il "professionista" del recupero, colui che non lo fa per interesse ma perché semplicemente vuole allargare la sua cerchia di amici e conoscenti, condizione indispensabile al reperimento di nuovi pezzi utili al completamento della sua collezione presente o futura.

In certi mercatini ci sono proprio quelli che ti vogliono fregare, come il tizio che frequenta Marzaglia, fuma come un turco e viene chiamato "il ladrone". I suoi prezzi sono esosi ma ti assicura che tutto funziona e che se il prezzo non ti va bene basta che tu lasci giù il pezzo che hai adocchiato. Non lasciatevi illudere dal fatto che pagando dieci Euro una scheda grafica ben esposta sul banco, invece di due Euro per dieci schede buttate alla rinfusa in uno scatolone, il funzionamento della prima sia garantito! Anzi, verrebbe da dire che avete maggiori probabilità di aver speso bene i vostri soldi affidandovi alla fortuna, piuttosto che finanziare certe persone, fossero anche solo dieci Euro!

Buona caccia quindi e ... attenti alle sòle!

[Tn]

Come eravamo...

Prima dei primi home (parte 2)

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Riprendiamo da dove ci eravamo interrotti sul precedete fascicolo per scoprire altri home disponibili prima del boom dell'inizio anni '80. Lasciato alle spalle il 1973 e gli incerti primi esperimenti di assemblaggio di un calcolatore personale, si stanno delineando delle vere e proprie linee guida per la progettazione dei microcalcolatori. Basso costo, modularità, espandibilità e dotazione software sembrano essere le prerogative di questi progetti che possiamo definire "seconda serie".

Mark-8

Questo progetto si deve alla rivista Radio Electronics (luglio 1974) che supportò e vendette un kit per costruire un home basato sul micro 8008 della Intel. Autore del progetto e degli articoli apparsi sulla rivista fu un certo Jon Titus che si offrì di aiutare quanti avessero avuto problemi nell'assemblaggio delle varie parti. Qualche anno dopo in una intervista dichiarò che erano stati venduti circa 400 Kit completi e oltre 7500 progetti per la costruzione in casa del sistema in tutte le sue componenti.

Lo stesso Titus ammette che la costruzione della macchina non era proprio alla portata di tutti gli hobbisti e che probabilmente la maggior parte di quanti hanno tentato di cimentarsi si sono arresi prima di arrivare alla conclusione. Una stima accettabile del numero di sistemi costruiti e che hanno funzionato si aggira fra le 1000 e le 2000 unità.

La guerra editoriale era iniziata e sei mesi più tardi (gennaio 1975) la concorrente rivista Popular Electronics ospitava la presentazione dell'Altair 8800, un sistema che usava la più potente CPU 8080 e che veniva offerto sia assemblato che in kit completo di tutte le necessarie com-



Il Mark-8 assemblato in un case "casalingo"

ponenti. Questo evento è famoso perché fu proprio quel numero di Popular Electronics ad ispirare Bill Gates verso i suoi primi passi nell'industria del software. Il destino del Mark-8 era inevitabilmente segnato. Attualmente si stima che di funzionanti non ne esistano più di una decina al mondo.

TV Typewriter II

Parliamo del TV Typewriter II, come di un personal da collegare al TV domestico, primo esempio di questa soluzione. In realtà era stato preceduto prima da una periferica con lo stesso nome (ora chiamata TV Typewriter I), realizzata su progetto di un certo Robert Sunding e progettata per diventare il terminale di output per il micro Mark-8 della rivista Radio Electronics.

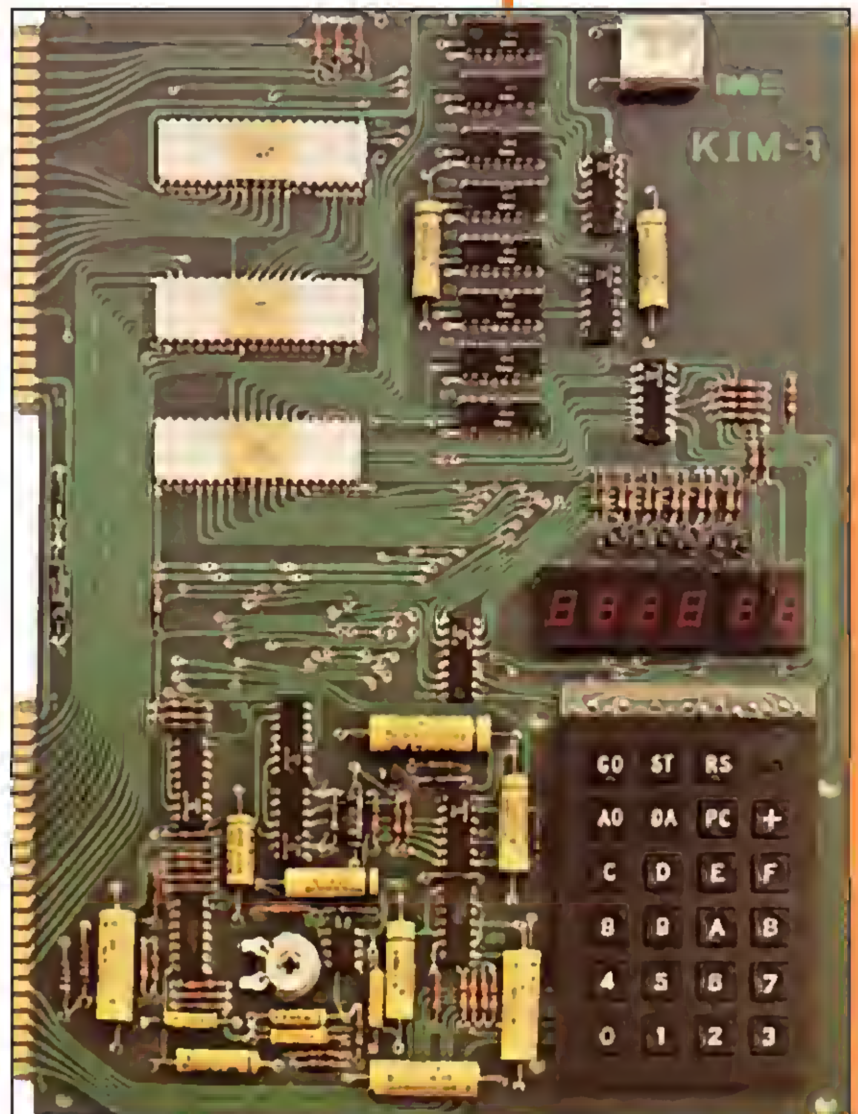
Successe in pratica che il successo del progetto Mark-8 (poche centinaia di esemplari, ma un numero significativo per l'epoca) desse il via alla nascita di uno user group di utenti, denominato "Digital Group" con tanto di newsletter e scambio conoscenze ed esperienze tecniche. Uno dei primi prodotti offerti all'attenzione dei soci (il diventato celebre "Packet #1") era una scheda video con 256 byte di RAM a bordo, in grado di pilotare un televisore e interfacciabile con il Mark-8 attraverso il bus di connessione delle schede. La scheda video non aveva a bordo alcuna logica di funzionamento e tutto doveva essere gestito via software, cursore compreso.

Dal funzionamento impossibilmente lento per la nostra percezione odierna, la TypeWriter è comunque degna di nota per l'idea innovativa che portava in sè, cioè l'utilizzo di un televisore per realizzare il terminale dei sistemi di calcolo personali.

KIM-1

Il successo nelle applicazioni dei primi microprocessori indusse le aziende concorrenti di Intel ad accellerare i loro progetti nel campo. Una di queste era la MOS Technology che lanciò il suo MOS6502

La scheda-kit denominata KIM-1. Uno dei progetti di grande successo nell'epoca pre-home.



(poi chiamato semplicemente 6502 quando si aggregarono altre source). Era il 1976 e il sistema di sviluppo a supporto era un vero e proprio microcalcolatore, chiamato KIM-1, annunciato con un redazionale sulla rivista Byte nell'aprile 1976.

L'intento dell'azienda era da un lato dimostrare le potenzialità del nuovo chip (che erano effettivamente molto buone) e dall'altro fornire un sistema educational per i tecnici che dovevano impararne l'utilizzo e la programmazione. Il suo prezzo contenuto (appena 245 dollari) fece breccia in un mercato insospettato, cioè quello degli hobbisti e il KIM-1 divenne un prodotto di grande successo, un best-seller si direbbe oggi.

La somiglianza del processore con il cugino Motorola 6800 e il prezzo davvero stracciato (appena 20 dollari quando il 6800 ne costava 200), determinarono l'attenzione delle aziende che si stavano cimentando con la nascita del mercato della microinformatica (Apple in primis), al punto che il chip divenne molto popolare e rimase quale cuore dei sistemi di calcolo home fino all'avvento dei 16 bit e oltre.

Il sistema è molto semplice ed è costituito in pratica da un'unica piastra, venduta senza alcun cabinet, che ospita oltre l'elettronica anche sei display a sette segmenti e una tastiera esadecimale. Fa parte del progetto un software "monitor" in ROM dotato di complete funzionalità per soddisfare le sperimentazioni degli utenti. Infatti è possibile collegare un terminale o una telescrivente o un registratore a cassette, il tutto pilotabile dalle routines in ROM. Il KIM-1 può essere quindi immediatamente utilizzato senza necessità di collegare alcuna periferica se non lo si desidera, inoltre la presenza della tastiera costituisce un passo avanti rispetto agli scomodi switch dei progetti precedenti.

Già all'epoca dell'introduzione del KIM-1 la MOS Tecnology era nel mirino della Comodore Computer per l'acquisizione, cosa se si realizzò nello stesso anno 1976 de-



Una foto dell'Altair 8800.

terminando anche il cambio di proprietà del KIM-1. Oggi i sistemi venduti originariamente dalla MOS e marchiati come tali sono considerati rari e più preziosi dei corrispondenti commercializzati dalla Commodore. E' notorio infine che proprio dal KIM-1 la Commodore prese il know-how necessario alla messa a punto del suo primo personal computer: il CBM.

Altair 8800

Nel 1975 l'ingegnere elettronico Ed Roberts della MITS, progettò il sistema Altair 8800, destinato a segnare una tappa fondamentale nella storia della conquista dell'elaborazione personale.

L'Altair ha lo stesso cabinet di quelli che allora erano considerati i calcolatori più abbordabili: i mini computer, era ciè assemblato in un case di dimensioni (19" x 19" x 10") contenente una motherboard con il Processore 8008 della Intel, 16 slot di espansione, un Front Panel con switch e luci di segnalazione e l'alimentatore. Il costo in kit era di 429 dollari e 621 se assemblato. A questi si devono aggiungere i costi per la memoria aggiuntiva (1 Kb per 97 dollari) e una periferica di I/O. Questa può scegliersi fra una telescrivente (la MITS la vende a 1500 dollari) o un terminale CRT dal prezzo più contenuto ma comunque "importante" di 760 dollari.

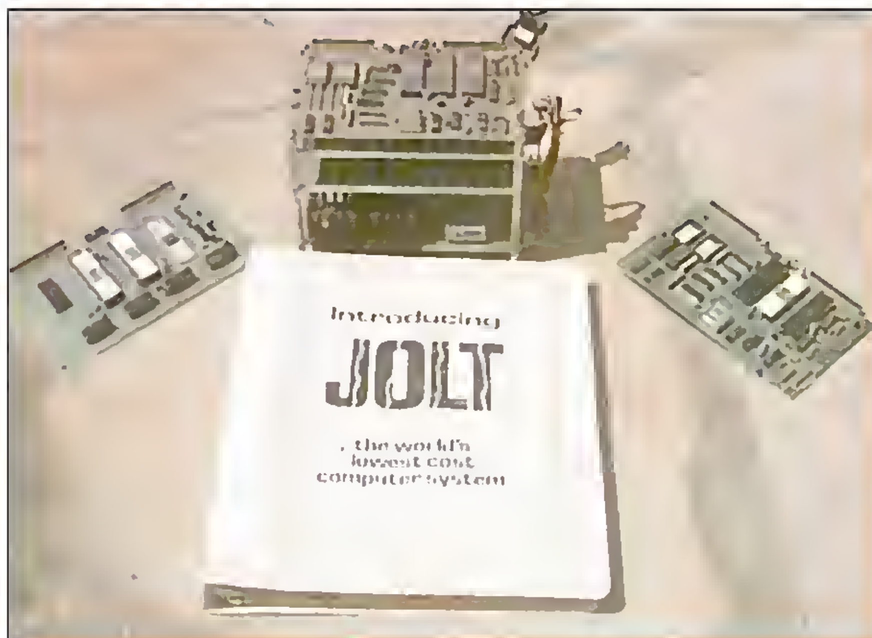
In definitiva un sistema "usabile" veniva sui 1900 dollari, il che non era certo poco per un hobbista, ma nonostante questo la MITS è riuscita a vendere parecchi kit prima che il suo prodotto fosse dichiarato superato.

Sphere 1

Progettato da Mike Wise, lo Sphere 1 era un sistema tutto-in-uno costruito attorno ad una CPU Motorola 6800. Dato che il chip era costoso, il kit risultante con 4 Kb di memoria arrivava 1400 dollari (versione assemblata). La potenza del processore 6800 si faceva però sentire, tanto che la Sphere poteva vantarsi di riuscire ad offrire delle periferiche davvero innovative per l'epoca, come i driver per floppy da 8", anche se il prezzo della doppia unità era fuori della portata di qualsiasi hobbista: 7995 dollari!

STP 6800

Ecco un'altro sistema basato sul microprocessore di Motorola. La Southwest Technical Products lo costruisce attorno ad una versione dimezzata del bus S-100 (chiamandolo S-50, che fantasia!). Contiene un boot-loader-monitor chiamato Mikbug nella ROM di sistema (uno dei primi esempi in questo senso). Con 2 Kb di RAM e l'interfaccia per un terminale veniva a costare in kit solo 450 dollari e in più la società commercializzava una espansione da 175 dollari per usare il televisore come display. Il progetto si deve



Micro computer Jolt

ad uno staff capitanato da Dan Meyer.

Fra l'altro la SWTPC è una delle poche compagnie produttrici di micro negli anni '70 ad essere ancora presente. Il suo business si articolò successivamente a quel primo progetto sulla realizzione di sistemi e controller basati sulla gamma di CPU della famiglia 68xx fino al 68000.

Jupiter

Si conosce pocchissimo di questo sistema; un po' più conosciuto il successore (Jupiter II) che è basato su un 6800.

Mike-2

E' un kit/assemblato attorno al processore Intel 8008 rilasciato dalla Martin Research nel 1974. Il prezzo del kit assemblato era di 345 dollari (100 in meno non assemblato).

RGS 008A

Costruito dalla RGS Electronics è stato introdotto nel settembre 1974 al prezzo (assemblato) di 375 dollari.

Jolt

Il Jolt è un classico 6502 assemblato sulla falsariga del successo del KIM-1, dalla Microcomputer Associates. E' famoso perché citato in un bollettino della NASA[1] dove veniva usato per l'elaborazione automatica di dati ricevuti da un "Omega Navigation System" (non chiedetemi cos'è 'sta cosa).

Tecnicamente si tratta di un kit che viene venduto a 249 dollari, definito dalla pubblicità "il computer in kit più economico del mondo"; il prezzo si riferisce però alla sola scheda CPU, poi bisogna comprare l'alimentatore la scheda I/O se si vuole gestire una TTY, la RAM per i programmi un po' più complessi di quanto sia possibile accomodare nella striminzita dotazione di base, etc... La facilità di montaggio è enfattizzata dalla dichiarazione che è possibile assemblarlo e interagire con la macchina in meno di tre ore.

Un'altra caratteristica, venduta per innivativa, è la presenza di un debugger/monitor denominato DEMON in grado di assemblare i programmi, gestire le periferiche e interagire con l'utente attraverso un terminale TTY.

Conclusione

Abbiamo visto in questa, per forza di cose incompleta, storia della nascita della micro informatica, come due fatti essenziali si possono assurgere a gemmatori di quella che diverrà negli anni '80 l'industria dell'home computer. I due fatti sono: l'intuizione di Intel che la guidò alla progettazione dei microprocessori prima e all'offerta del kit di sviluppo SIM4-01 prima e SIM8-01 poi e come secondo evento il proliferare di offerte di scatole di montaggio che occupano gli anni dal 1973 al 1978.

Questi kit hanno tutti più o meno le stesse caratteristiche e alla fine, salvo qualche eccezione, costano più o meno uguale. Una CPU, 1 Kb di RAM e al massimo 1 Kb di ROM formano la dotazione elettronica di base. A questa si aggiungono le interfacce per telescrivente e registratore a nastro perforato. Terminali CRT e floppy sono un lusso mentre fa capolino la periferica "principe" degli home anni '80, cioè il registratore a cassette.

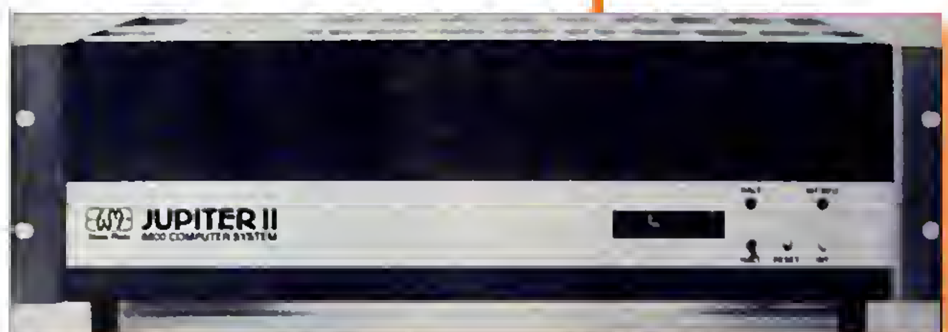
Il mercato dei kit ha un suo ultimo sussulto con l'avvento dei microprocessori ad 8 bit di ultima generazione come l'Intel 8080, lo Zilog Z80, le CPU di Motorola e il 6502 della MOS Technology. Ma ormai la strada dell'autocostruzione era preclusa e premevano i grandi assemblatori come Apple, Commodore, etc... che avevano bisogno di vendere molto e quindi non potevano accontentarsi dei soli hobbisti. Qui comincia un'altra epoca, quella degli home veri e propri e del software in dotazione. Forse proprio l'esplosione del software, per molti versi inaspettata, ha contribuito in maniera determinante all'imposizione dell'informatica personale nella società moderna.

[Tn]

Bibliografia

[1] Lilley, R. W. - "Demonstration program for Omega receiver prototype microcomputer data processing" - NASA technical Report 1976.

[2] Zbigniew Stachniak - "Intel SIM8-01: A Proto-PC" - IEEE Annals of the History of Computing - gennaio-marzo 2007.



Jupiter II

Le prove di Jurassic News



Olivetti M10

Una combinazione di tecnologia nipponica e design italiano in questo prodotto della casa di Ivrea, presentato nel 1984 come "ufficio trasportabile".



In apertura una pubblicità Olivetti che presenta l'M10 come il perfetto ufficio da viaggio.

Introduzione

Quando sono andato a riprendere il mio M10 dalla scatola di cartone dove l'avevo deposto nel mio personale magazzino al riparo dalla polvere, allo scopo di realizzare questo articolo, mi è preso un senso di acuta nostalgia. La mente è ritornata allora all'epoca in cui un mio zio, maresciallo dei Carabinieri, me l'aveva donato conoscendo la mia passione per l'informatica.

Non era nuovo il gioiellino Olivetti ricevuto così proficuamente, e una vistosa "cicatrice" denunciava un "colpo basso" subito forse in una caduta o chissà che altro genere di sventura. Mio zio mi disse semplicemente che veniva da un sequestro e, come molta merce, non più reclamato dal legittimo proprietario e quindi alienato (il che significa che era stato messo all'asta e mio zio l'aveva riscattato per poche lire, non posso dire quante).

Caro, simpatico e generoso zio che non mancava mai di portarmi un regalo ogni volta che "tornava alle origini", come diceva lui, lasciando la sua Firenze, diventata patria d'adozione, per le natie montagne del Trentino.

L'M10 è stato un regalo proprio bello, direi eccezionale! Seguito, lo ricordo bene, da una stampante ad aghi (NEC se non ricordo male) e qualche anno dopo da un M24, sempre Olivetti. Grazie anche a questi generosi e azzeccati regali sono stato in grado di essere spettatore della storia dell'informatica fin dalle sue origini.

Purtroppo sono quasi vent'anni che zio Alberto non è più e quasi altrettanti che l'M10 ha smesso di funzionare, o meglio: che il display ha smesso di funzionare, perché sono sicuro che il resto della macchina è a posto. Per un po' ho

cercato un ricambio ma, mancate alcune occasioni per esosità del venditore e assodato che proprio il display è la componente più fragile della macchina, ormai mi sono rassegnato a rimandare la riparazione ad una improbabile (per quanto possibile) scoperta fortuita.

Lo stesso destino, in fondo, di un sacco di altre cianfrusaglie informatiche e non, delle quali non ho voluto disfarmi, un po' per ragioni affettive e un po' per la mia indole alla conservazione. Se non potrò io, chissà che qualcun altro, magari con il display a posto ma con la motherboard guasta, non possa un giorno ridare vita a uno dei primi portatili apparsi sul mercato.

La mia recensione non è basata su prove dirette, non potendo utilizzare il sistema per le ragioni che ho detto, ma su quanto mi ricordo e sulla documentazione in mio possesso, oltre che grazie alla onnipresente Internet!

Primo approccio

Il portatile della Olivetti (perché di un vero portatile si tratta), nasce attorno al 1983 da un progetto giapponese della Kyocera che lo commercializzò coin il nome di KC-85, acquistato e customizzato dalla ditta di Ivrea come anche da altri costruttori di personal dell'epoca. In Italia appare nel primo trimestre del 1984, presentato con grande curiosità ed enfasi dalle riviste del settore.

Figli dello stesso progetto sono una serie di altri sistemi, come ad esempio il Tandy-100, notevolmente diffuso negli States. Tutti questi sistemi condividono l'hardware e il software; si differenziano per le soluzioni meccaniche e di design. Da questo punto di vista l'M10 si stacca dal piattume di un oggetto rettangolare, grazie al display mo-



L'immagine valorizza il design della macchina con il display in posizione di lavoro.



Una bella immagine del sistema in assetto di utilizzo.

bile che si inclina a favorire la visibilità.

Nel 1983/84 il mercato richiedeva evidentemente una tecnologia trasportabile, segno inequivocabile che l'elaborazione personale delle informazioni stava diventando una commodity uscendo da quell'aurea di tecnologia da seminterrato cui era stata relegata per tanto tempo a causa delle origini umili (chi può negarlo) del personal e della gestione dell'informazione elettronica in generale.

Il risultato dell'adattamento eseguito in Italia (o per conto di, non possiamo saperlo), dimostra il genio italiano e la qualità del design della Olivetti che è riuscita a costruire un oggetto bello, oltre che funzionale. Si potrebbe dire, senza ombra di smentita, che la Olivetti è stata per l'Europa quello che la Apple è stata per gli States: la fantasia al potere.

L'M10 si distingue al primo colpo da qualsiasi altro "portatile" dell'epoca e questo grazie al particolare colore "oro satinato" della cornice che ospita il display mobile che si inclina verso l'utilizzatore e dalle vistose scritte "Olivetti" e "M 10" che risaltano sulla destra.

Sotto il display una "vera" tastiera Querty che non fa affatto rimpiangere un analogo dispositivo per PC desktop.

Accoppiata tastiera-display (nonostante le limitazioni di quest'ultimo) fanno capire perfettamente che ci si trova difronte ad un oggetto che non lascia indifferenti.

Visto con gli occhi di oggi non si potrebbe dare un voto alto all'apparecchiatura, soprattutto perché il display non è particolarmente brillante e va ad occupare uno spazio limitato rispetto alla dimensione che sarebbe disponibile. Ne risulta una cornice piuttosto massiccia e che poteva forse essere meglio sfruttata, ma per l'epoca ci si può accontentare...

Un uso spinto della tecnologia CMOS e il display LCD, permettono al sistema una discreta autonomia nell'alimentazione a batterie e una completa dotazione di interfacce ne permettono il collegamento alle periferiche più comuni: stampante parallela e dispositivi seriali come ad esempio un modem o un accoppiatore acustico. Ne esiste una versione con modem integrato (assente nel mio esemplare), forse per lo scarso interesse per la telematica in Italia alla metà degli anni '80.

Il computer non è poi tanto piccolo (30 cm di larghezza x 22 cm di profondità) per cui non si discosta troppo dalle dimensioni dei portatili di oggi. Quello che lo differenzia è semmai lo spessore: quasi 6 cm per un peso che è poco inferiore ai due kilogrammi.

Il display è reclinabile, come si diceva, ma non troppo: nel senso che non lo di può posizionare in verticale ad esempio; mentre in posizione di riposo rientra nel profilo del corpo macchina facilitandone il trasporto in una qualsiasi borsa o ventiquattrore senza troppi problemi.

La visibilità la ricordo abbastanza buona, certo facilitata dalla possibilità di inclinare il display e l'uso della tastiera dà un discreto feedback. Peccato per lo spessore del corpo macchina che non rende del tutto agevole la digitazione; ma sono particolari non troppo importanti per questa classe di sistemi.

L'M10 nasce con la vocazione "on the road", di essere cioè una macchina da viaggio e da questo punto di vista risulta un progetto azzeccato e quasi ideale. La dotazione di porte è completa ed è dotato di una innovativa modalità "RAM permanente" che ne consente l'uso come memoria di massa.

Il sistema si accende tramite un interruttore a slitta alloggiato sul fondo del sistema. Un secondo interruttore dello stesso tipo ma meno accessibile, attiva la memoria permanente. Sul retro in buon ordine guardando da sinistra: il pulsante di Reset, il jack per l'alimentatore esterno, la seriale a 25 pin, la parallela (purtroppo non Cannon ma proprietaria), il classico DIN per il registratore a cassette e una strana porta a nove pin siglata BCR che si scopre dal manuale essere predisposta per un lettore di codici a barre.

Nel mio modello, ma credo in tutti i modelli commercializzati in Italia, manca l'uscita del modem interno





Il manuale (versione inglese), allegato al prodotto e l'immancabile monografia Jackson



Il retro a grandezza naturale (o quasi).

che troverebbe posto vicino al connettore del registratore.

Sul fondo della macchina, oltre ai due citati interruttori, troviamo il vano porta pile (quattro AA da 1,5 Volt) che forniscono la tensione di 6 Volt necessaria in mancanza dell'alimentatore esterno; uno sportellino nasconde l'espansione del bus di sistema (non mi sembra di aver mai avuto notizia di quache espansione in tal senso), mentre un'altro sportellino accede agli slot di espansione della RAM.

La documentazione è in linea con l'epoca di commercializzazione, con grande enfasi sulla tecnologia e ricchezza di particolari tecnici (peraltro indispensabili al funzionamento di alcune opzioni). Particolarità del collegamento alle banche dati e descrizione di come modificare un cavo seriale per adattarlo alle bizze di una sconosciuta periferica, rendono l'atmosfera di un tempo che fu ed è ormai lontano.

Il processore OKY che equipaggia il personal M10.



Hardware

Il progetto è costruito attorno ad un processore di fabbricazione OKI siglato 80C85, un 8 bit in tecnologia a basso consumo che "viaggia" a 2,4 MHz.

Il processore OKY 80C85 è compatibile con la serie 8085 di Intel. Si tratta di una "normale" evoluzione dell'architettura 8080, ormai un classico micro a 8 bit con 16 linee di indirizzo per un totale di 64 Kb indirizzabili e 256 porte di I/O.

La differenza fra la famiglia 8080 e la 8085 sta nella maggiore integrazione funzionale del chip 8085, che include ad esempio i circuiti di clock, nella presenza del bus multiplexato fra dati e indirizzi (parte bassa), nella specializzazione degli interrupt e delle linee di controllo dell'I/O e in una semplificazione circuitale (prevede ad esempio un'unica tensione di alimentazione a +5V rispetto alle tre tensioni necessarie all'8080.

La dotazione minima di RAM è di 8 Kb ma si espande a colpi di 8 Kb fino al massimo di 32 Kb. La ROM è invece da subito sostanziosa: 32 Kb che possono diventare 64 Kb (recita il manuale, ma a vedere la mappa di memoria non si capisce come), immagino utilizzando il bus



di espansione o gli slot disponibili.

Lo spazio di indirizzamento, che dispone di 65000 indirizzi, è equamente diviso fra ROM nella parte bassa (i primi 32K) e RAM nella parte alta, dove trova spazio la memoria di lavoro e quella "video" (se così si può chiamare il buffer dell'LCD). Ovviamente i buffer si mangiano un pezzetto di RAM, per quanto limitato sottraendola alla disponibilità dell'utente.

La commercializzazione prevedeva alcuni modelli che si differenziano per tre caratteristiche fondamentali offerte in varie combinazioni: tastiera italiana o USA, dotazione di memoria dal minimo di 8 Kb fino a 24 Kb on-board ed infine presenza o assenza del modem integrato.

Il display LCD permette la visualizzazione di 8 righe da 40 caratteri ciascuna; i caratteri sono di discreta dimensione ma al solito, le righe sono ben poco spaziate. Sono previste le minuscole e una discreta dotazione di segni e simboli semigrafici. La modalità grafica permette di indirizzare tutti i pixel singolarmente realizzando una matrice di 240x64 punti.

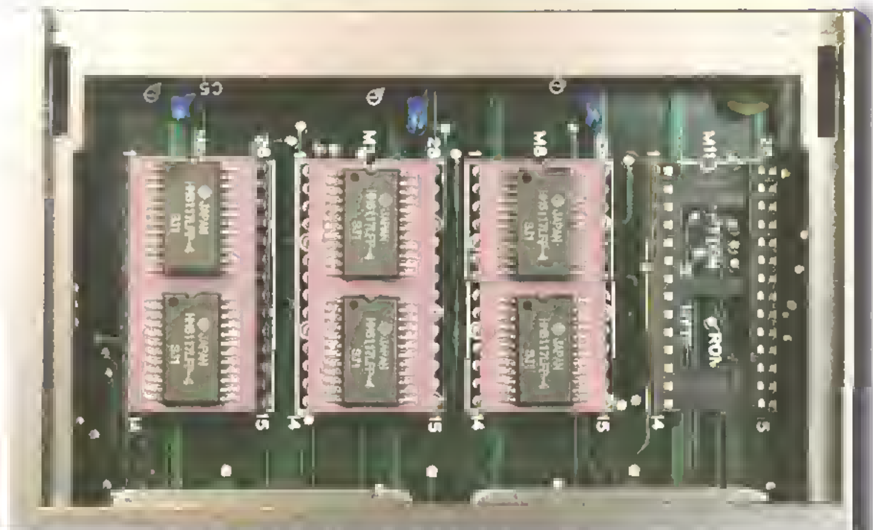
L'inclinazione del display è ottenuta con una cerniera che nasconde una molla all'interno. Dopo qualche tempo il meccanismo si irrigidisce un pochino e l'azione di sollevare il display è accompagnata da quello che molti hanno definito "il gemito nell'M10".

La tastiera è meccanica con buon feedback ed appare robusta e ben proporzionata. La dimensione dell'unità ne permette una adozione a dimensione standard con tastierino numerico "annegato", come oggi succede nei laptop.

Per il suono l'M10 si affida ad un generatore interno da 5 ottave e ad un piccolo altoparlante nascosto nel corpo macchina.

L'interfaccia registratore ha sollevato molte critiche, ricordo, per una presunta inaffidabilità. Personalmente non mi sembra di avere avuto questi problemi, o almeno, di non averne avuti di aggiuntivi rispetto alle altre macchine. Certo è meglio usare una unità a cassette

Gli slot per l'espansione della RAM/ROM



di buona qualità: ricordo in particolare un registratore Philips che ha fatto il giro di tutti i miei home, adattandosi senza troppa difficoltà. Se il registratore dispone di REMOTE, esistono i comandi di accensione e spegnimento del motore accessibili da BASIC.

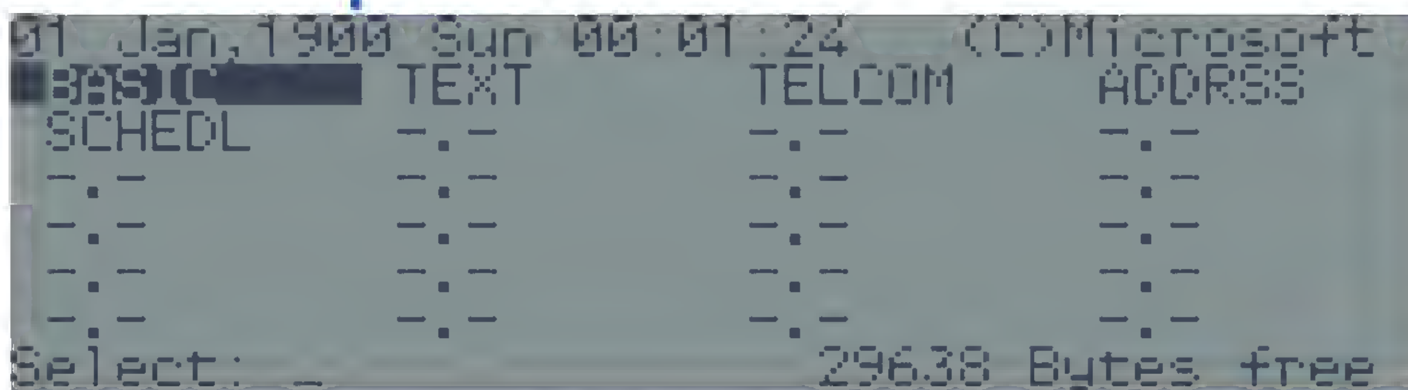
L'architettura è quanto di più semplice si possa immaginare: CPU centrale che pilota RAM e ROM sul bus e una unità di controllo dell'I/O che si occupa di tutta la parte comunicazione, tastiera e display compresi. Per ogni periferica la Olivetti offre il cavo o la combinazione di cavi adatti allo scopo. Come periferica è degna di nota una printer/plotter parallela dalle dimensioni contenute in linea con il sistema centrale.

Grazie alla tecnologia CMOS e al display LCD, una dotazione di pile stilo (4 AA per la precisione), assicura una autonomia di circa 40 ore, praticamente una intera settimana di lavoro: davvero non male! Il livello di bassa carica delle batterie è segnalato da un led posto sulla cornice del display.

Uso

Quello che rendo diversa questa macchina dagli home o office computer dell'epoca è sicuramente la trasportabilità. Essa si realizza non solo nella compattezza e omnicomprensività della dotazione ma anche e soprattutto nella presenza "embedded" di una suite di applicativi adatti ai lavori più comuni: interprete BASIC, editor di testo, rubrica indirizzi, agenda elettronica e programma di comunicazione.

Il sistema dispone di un rudimentale sistema operativo che prende il controllo della macchina all'accensione e consente l'esecuzione dei programmi, la gestione delle periferiche (ad esempio il caricamento da cassetta) e la navigazione nella "directory". In realtà non è proprio un file-system come oggi siamo abituati ad usare, con cartelle, sottocartelle, etc… Qui si parla semplicemente di un unico sapzio di memorizzazione con unalista di file da selezionare. I file hanno un nome e una estensione di due caratteri che ne stabilisce anche il contenuto. Ad esempio .DO è riservata ai documenti, .BA sono i sorgenti BASIC, etc…



Il menù all'accensione (il display non è reale, è stato ricostruito per la difficoltà di ripresa).

Per eseguire uno dei programmi, ad esempio il BASIC in dotazione, basta posizionarsi sul nome (che viene evidenziato in reverse) con i tasti freccia e premere INVIO, oppure si può direttamente scrivere il nome del programma e confermare.

Sul display è sempre presente una riga informativa (la prima in alto) con data e ora, mentre nell'ultima riga vengono indicati i byte di memoria disponibili.

Eseguito il BASIC ci si trova con il classico prompt Microsoft: "OK", in stato "comandi". Lo stato complementare è quello di esecuzione quando viene lanciato il progrmma in esecuzione.

Dal momento che la memoria RAM funge anche da storage, esistono dei comandi che realizzano una rudimentale gestione dei file che possono essere al massimo 19 (d'altra parte con 24 Kbyte di RAM non si possono fare miracoli). Si possono cancellare file non più usati con KILL "nome" o rinominare con:

```
NAME "MIOFILE.DO"
      AS "TUOFILE.DO"
```

Il comando IPL consente di nominare un programma all'autoesecuzione al momento dell'accensione della macchina:

IPL "BASIC"

Predispone all'esecuzione dell'interprete ad ogni accensione.

Si comincia a parlare di integrazione dei programmi con il BASIC dotato di istruzioni per interagire con i file prodotti dagli altri programmi e con l'editor di testo da usare anche per la stesura dei sorgenti. Quest'ultima una vera evoluzione rispetto alle limitate capacità di editing degli interpreti visti finora.

Grazie alla ROM e alla memoria RAM non volatile, il computer si accende praticamente istantaneamente mostrando il menù principale nel quale si naviga scegliendo l'applicazione da eseguire fra quelle in dotazione o quelle create in proprio. Esiste anche la possibilità di un "autoload" con lancio di una particolare applicazione al momento dell'accensione.

Il prompt del BASIC con la quantità di memoria RAM disponibile.

```
OLIVETTI M10 BASIC 1.0
(C) 1983 Microsoft
29381 Bytes free
Ok
```

Il BASIC

Il BASIC è un classico Microsoft, e non ci sembra il caso qui di passarne in rassegna le istruzioni. Accenniamo solo a quelle che sono direttamente legate al sistema M10.

```
DATE$ = "12/02/08"
```

Queste funzionano sia con effetto immediato sui parametri di funzionamento della macchina e sia come funzioni richiamabili da BASIC. Ad esempio:

```
10 PRINT TIMES$
```

stampa l'ora attuale nel formato hh:mm:ss

Interessante la possibilità di programmare l'esecuzione ad orario. Ad esempio:

```
10 TIME$ ON
20 ON TIME$="12:00" GOSUB 100
...
100 REM - suona il rancio
120 BEEP:BEEP
130 PRINT "Il pranzo e' servito"
```

L'uscita dal BASIC si ottiene con con il classico "system" ma con il comando "MENU".

Salvataggio e caricamento di programmi non si effettua dall'interprete ma attraverso il sistema operativo che sfrutta una parte di RAM come buffer o "memoria di lavoro", come viene chiamata sul manuale.

Ad esempio:

`CLOAD "PROVA"` *- carica il programma "PROVA" dalla cassetta alla memoria di lavoro;*

`CLOAD "PROVA", R` *– carica il programma "PROVA" dalla cassetta e lo manda in esecuzione*

Con il comando LOAD è possibile specificare la cassetta (estensione "CAS") o la RAM (estensione RAM) come destinazione del caricamento. Alcuni esempi:

```
LOAD "CAS:FILE1"
LOAD "RAM:FILE2"
LOAD "COM:FILE3"
```

Il terzo esempio coinvolge la porta di comunicazione seriale.

Utile e innovativa anche l'istruzione POWER che, come si capisce dal nome, si occupa dell'accensione e spegnimento programmato della macchina. E' una istruzione con molte possibilità, vediamo un piccolo esempio:

```
10 PRINT "INIZIO DEL MIO PROGRAMMA"
20 POWER OFF, RESUME
30 PRINT "CONTINUAZIONE ALLA RIACCENSIONE"
40 STOP
```

Questo breve sorgente stampa il primo messaggio e poi spegne il calcolatore. Quando esso viene riacceso il programma riparte tranquillamente dalla riga 30.

Per quanto rudimentale l'M10 si picca di gestire un minimo di suono con l'istruzione SOUND. I parametri sono due: frequenza e durata della nota.

I programmi in dotazione

Data l'epoca di uscita di questo sistema, è più interessante occuparsi della suite di applicazioni disponibili nella ROM piuttosto che dell'arcinoto BASIC. L'idea di avere disponibile uno strumento di lavoro trasportabile comincia a farsi strada e ciò che rende l'M10 appetibile per una certa classe di utilizzatori è da ricercare nel software piuttosto che nelle capacità programmabili, pur interessanti, della macchina.

La suite applicativa comprende un editor di testi (TEXT), una indirizzario (ADDRSS), un programma di comunicazione (TELCOM) e una agenda elettronica (SCHEDL). Manca come si vede un foglio di calcolo e questo è davvero un peccato; immaginiamo che sia stato sacrificato per mancanza di spazio nella RAM o forse perché all'epoca l'idea dello spreadsheet non era ancora nel DNA dell'utilizzatore (chi potrebbe fare oggi a meno di Excel?).

TEXT è poco più di un text editor che lavora esclusivamente in memoria (e quindi il testo deve trovare spazio nei byte disponibili). Utilizza il display come finestra sul buffer dei dati e dispone delle principali funzionalità che ci si aspetta di trovare in un simile prodotto. Gli otto tasti funzione della tastiera sono associati ad altrettante funzioni che vanno dalla ricerca con sostituzione del testo fino al cut / paste di una selezione per finire con F8 che chiude il programma (ma il testo è comunque presente in RAM grazie al fatto che giò all'apertura è necessario assegnargli un nome) e ritorna al MENU di sistema, pronto ad accettare l'esecuzione di un nuovo comando.

Grazie all'associazione estensione-programma, i file con estensione .DO, che viene automaticamente affibbiata al nome scelto per il documento, è sufficiente digitarle il nome per richiamare l'editing del testo contenuto.

Il programma ADDRSS è una semplice rubrica telefonica con indirizzario che consente di portarsi appresso una discreta quantità di contatti (dipende sempre da quanta RAM si ha a disposizione). La rubrica è in sè e per sè null'altro che un file di testo gestibile con TEXT; semplicemente il testo, formattato in una certa maniera, viene utilizzato da ADDRSS con dei comandi mirati.

Il formato del contatto prevede un record per riga con informazioni separate da due punti; qualcosa come:

Mario Rossi: 027=821998 : via della Repubblica....

Il primo campo (cioè il testo all'inizio della riga fino al primo ':' è considerato il nome del contatto e la parte sulla quale si eseguiranno le funzioni di ricerca. Se il secondo campo viene usato per il numero di telefono è anche possibile (con la versione dell'M10 con modem

integrato), effettuare direttamente la chiamata.

L'indirizzario funziona solo se il file con i dati si chiama ADRS. DO, pertanto se si vogliono gestire rubriche diverse o usare le funzionalità per realizzare semplici database sarà necessario rinominare il file interessato prima di usarlo.

Un file di testo con nome NOTE. DO è invece il contenitore delle informazioni per il programma SCHEDL. Anche in questo caso il file è semplicemente un testo preformattato e contenente un record per riga. SCHEDL semplicemente mette a disposizione delle semplici funzioni di ricerca e gestione.

TELCOM è l'applicativo più "innovativo" (siamo pur sempre nel 1983!) dell'intera suite. É curioso sfogliare il manuale per scoprire gli sforzi profusi nello spiegare i termini e il funzionamento di un programma di comunicazione. Data la novità di questo genere di applicativi, la parte dedicata al programma di comunicazione è più corposa delle altre. TELCOM lavora integrandosi con gli altri applicativi, ad esempio si può fare la chiamata utilizzando la registrazione del numero di telefono contenuta nell'indirizzario. In questo caso è possibile conservare nel record della rubrica, oltre al numero di telefono da chiamare, anche user id e password, oltre ai principali parametri di comunicazione.

Per il resto anche qui non c'è nulla di particolarmente interessante da riferire se non la presa d'atto di come le possibilità di comunicazione a distanza hanno compiuto i loro primi passi.

Come altri software al loro primo apparire, anche i programmi di comunicazione hanno dovuto sgomitare guadagnarsi una credibilità ed una usabilità che non prevedesse la presenza di un tecnico o di un utilizzatore smanettone. Anche TELCOM per esempio, richiede l'impostazione dei parametri di comunicazione, per cui si richiede all'utente di sapere cosa significa un bit di stop o il flag di controllo di parità. La velocità di comunicazione dipende evidentemente dal sistema chiamato; l'M10 arriva fino ai 19 Kboud: oltre era pura fantascienza per l'epoca.

Le funzionalità sono quelle standard: emulazione di terminale, up e download. Il terminale non emula nessun sistema esistente ma si limita alla gestione dell'ECHO video con una qualche funzionalità di scroll bidirenzionale della finestra di testo.

Conclusioni

Non si può che essere entusiasti di questo sistema completo e divertente nell'utilizzo. Immagino l'entusiasmo, pari al mio, dei primi possessori e l'incredulità di poter disporre di un compagno di lavoro così flessibile e utile!

Certo la tecnologia ha un costo: 2.400.000 Lire che non sono affatto poche per la versione base. La pulizia del cablaggio e la cura dei particolari, tipica della grande Olivetti (quella di un tempo, non quella che è venuta dopo :-) ne fanno ancora oggi un oggetto di culto per gli appassionati del vintage.

Bibliografia

Olivetti M10: guida all'uso; Massimo Mangia; Jackson editore; 1984

http://www.mainieri.it/WEB/La_raccolta_di_strumenti_di_calcolo/Pagine_singole_macchine/Olivetti_M10.htm

http://retrocomputing.altervista.org/m10/m10-storia.html

m&p computer n. 34

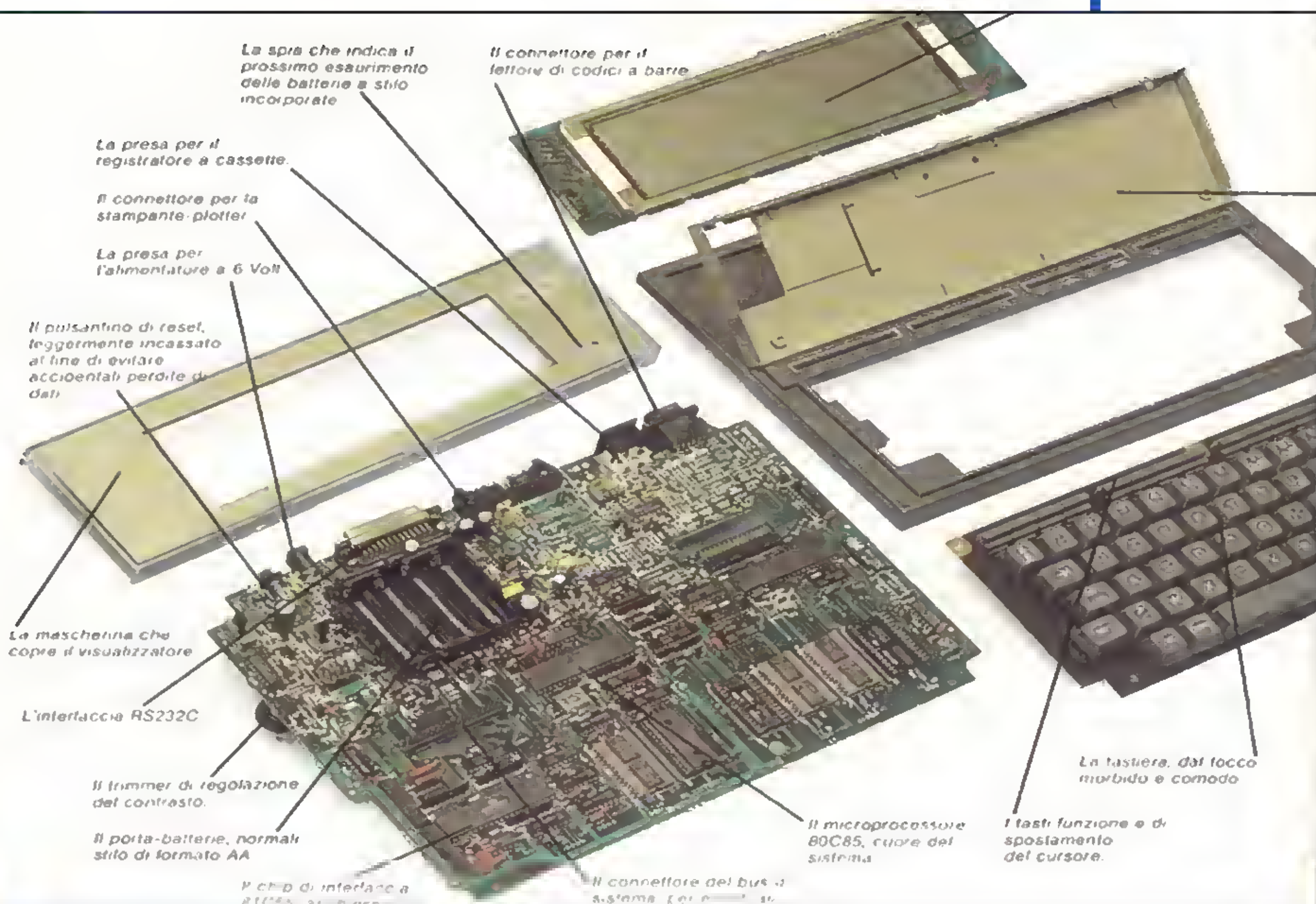
microcomputer n. 10

Bit n. 49

Chip n. 3

[Sn]

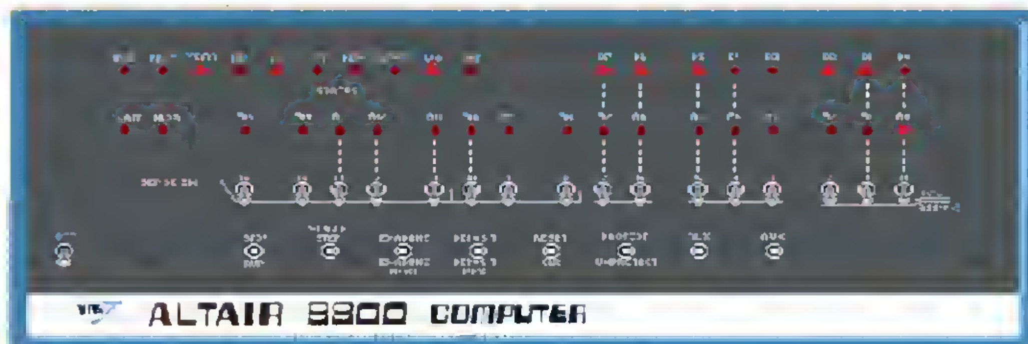
Una immagine dell'"esploso". Si noti l'estrema pulizia del progetto della piastra madre.



Emulazione

Altair32

I mondi virtuali a volte possono essere molto realistici...



In apertura il pannello frontale di un Altair 8800, da molti considerato il primo "vero" home computer della storia.

Introduzione

Tutti i retro-computeristi conoscono o almeno hanno sentito nominare l'Altair 8800, uno dei primi sistemi a microprocessore dal prezzo abbordabile agli hobbisti (solo statunitensi, è ovvio!), apparso nel 1975 e quindi prima dei vari Apple II, CBM, VIC20 etc...

Il suo debutto, nel gennaio 1975, è stabilito da un articolo apparso sulla rivista Popular Electronics a supporto della vendita dell'oggetto in kit per circa 400 dollari. La ditta che lo costruisce è la Micro Instrumentation and Telemetry Systems (MITS, da cui l'identificazione che a volte si trova di "MITS 8800").

Era quella una stagione (dal 1975 al 1980) che vedeva la nascita dell'industria dei micro computer e molte aziende, anche improvvisate, si cimentavano con la progettazione di un sistema di calcolo personale dedicato agli appassionati di elettronica. Contemporaneamente facevano capolino i primi software, linguaggi e utilities per la gestione di periferiche, strumenti indispensabili per il futuro della micro informatica. L'esplorazione, perché di questo si tratta, di quei proto-sistemi viene classificata come archeologia informatica anche se a ben vedere già negli anni '40 i sistemi di calcolo erano apparsi nei laboratori di ricerca e nelle organizzazioni aziendali.

Disporre oggi di un emulatore per il sistema Altair 8800 è una occasione eccezionale per tuffarsi in quegli anni ruggenti e questo articolo fa proprio questo. Siete pronti a saltare dal trampolino? Andiamo...

Il prodotto

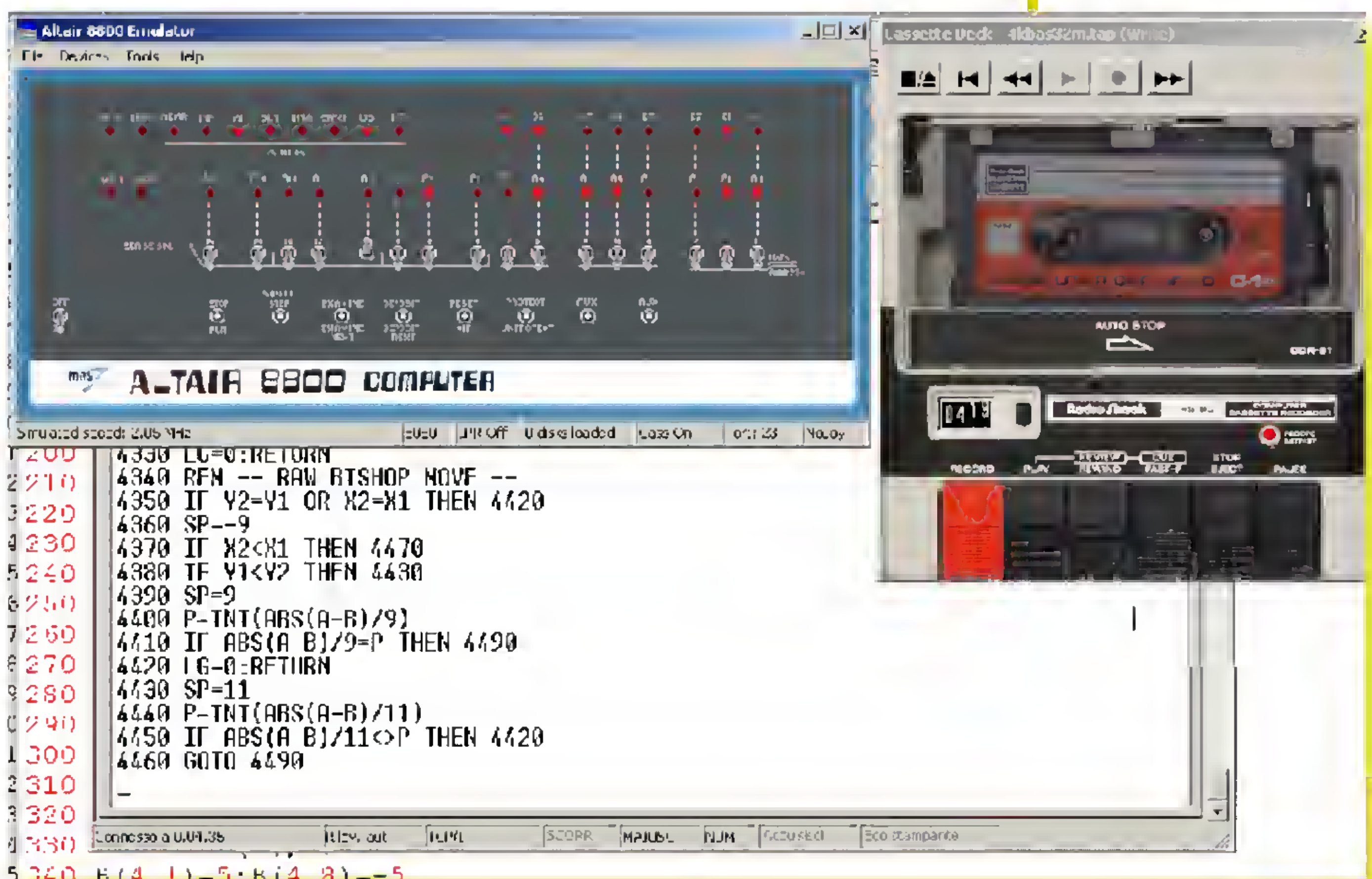
Altair32, questo il nome del prodotto, è un progetto che viene portato avanti dal 1997 con il contribu-

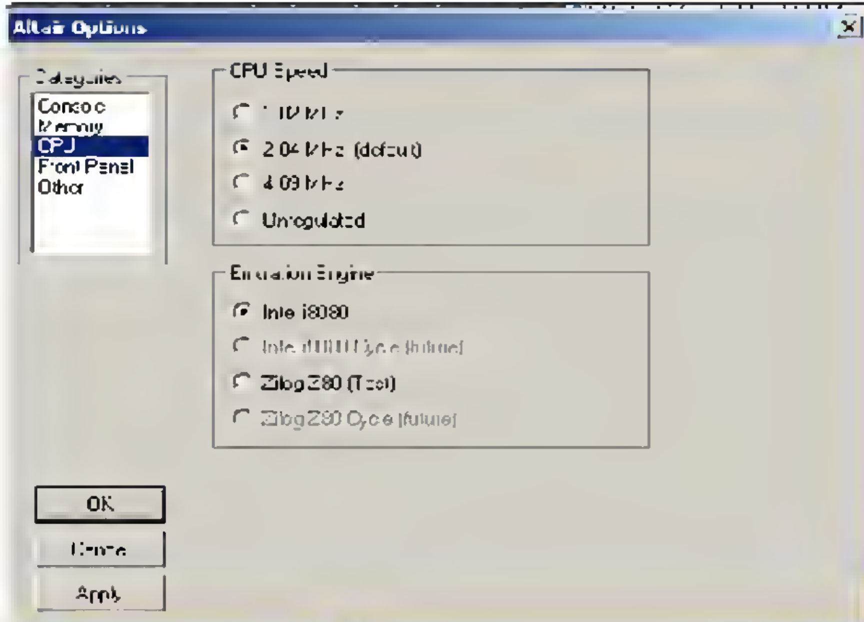
to di varie persone (una quindicina) ed assume l'aspetto di un classico progetto Open. L'emulazione è un campo applicativo che può dare molte soddisfazioni, per quanto difficile da dominare, e la messa a punto di un sistema richiede anni di sforzi, se pur distribuiti nel tempo, prima di diventare veramente operativo al 100%. Questo è un po' quello che accade per Altair32 che è ormai giunto ad un livello più che buono rendendo disponibile l'emulazione di periferiche e software anche abbastanza sofisticato, come ad esempio il sistema operativo CP/M e conseguente emulazione di floppy e hard disk.

Le specifiche tecniche del sistema originale in breve sono le seguenti: CPU Intel 8080, 4 Kb di RAM statica, due porte seriali RS232 (disponibili anche le meno usate Current Loop). La ROM di sistema varia a seconda delle versioni ed occupa la parte alta della memoria (4 Kb o 8 Kb), lasciando liberi 56Kb di indirizzi per eventuali espansioni di memoria. Il floppy è da 8 pollici hard-sectored, che significa che i settori sono individuati da altrettanti buchi sulla superfice del floppy, non uno solo come nella tecnologia soft-sectored che è più recente.

L'attuale versione dell'emulatore (siamo alla 3.20.0400 al momento della stesura di questo articolo) supporta una grande quantità di parametri e permette di configurare virtualmente un sistema Altair veramente completo. Il clock della CPU parte da 2 MHz, sono disponibili 60 Kb di RAM (nella configurazione con 4 Kb di ROM), otto controller floppy per un massimo di 16

L'emulatore è in funzione. Da cassetta abbiamo caricato il BASIC e stiamo provando a programmare...





unità in linea e c'è pure il supporto per volumi "grandi" possibili sotto CP/M, una uscita "terminale", una stampante "a linea", interfaccia per lettore di nastri perforati e una per il registratore a cassette.

Assieme all'emulatore arrivano i necessari file binary come le ROM nelle varie versioni e le immagini dei software più interessanti (il CP/M, il Basic, etc...).

Installazione

Il pacchetto disponibile per Windows 32 viene scompattato in una directory a piacere e questa operazione costituisce l'installazione (non c'è il classico setup.exe per capirci).

Mi permetto di fare una osservazione personale plaudendo alle installazioni che si configurano come semplice copia sul file system, senza pesanti interventi su registry o altro! Sono anche molto più comode le disinstallazioni e non si rischia di togliere dal sistema qualche DLL potenzialmente utilizzata da qualche altro prodotto.

Tornando al pacchetto, esso va ad occupare la bellezza di 25 megabyte, anche perchè contiene i sorgenti e tutto il necessario alla compilazione (che deve essere fatta con il GCC e non con il Visual C++ di Microsoft).

Presenti anche una discreta quantità di esempi, per la maggior parte in BASIC ma anche in Assembler, utili per studiare le intime caratteristiche dell'interprete, meglio sarebbe dire degli interpreti, forniti a corredo.

Uso

Lanciato l'eseguibile Altair32.exe, ci troviamo una bella immagine del pannello frontale in una delle tante versioni che variano più che altro per il colore. Il sistema è spento e il mouse si trasforma in una manina pronta ad operare sul pannello di comandi. Notiamo in basso la barra di stato che ci fornisce alcuni parametri di funzionamento: il tipo di CPU, lo stato dei dischi, cassetta e stampante. L'emulatore permette (ma solo sperimentalmente) di sostituire la CPU 8080 con lo Zilog Z80 che come si sà è perfettamente compatibile a livello di istruzioni binarie.

Una visita al pannello di configurazione accessibile da menù ci indica la configurazione in uso permettendo le eventuali modifiche del caso. Si possono anche avere

i suoni (meglio, i rumori) generati dalle componenti varie. La configurazione stessa viene salvata in un file .ini "vecchia maniera" presente nella stessa directory di prodotto.

L'attivazione dello switch on/off ci restituisce il sistema funzionante con alcuni led accesi. Si tratta ora di capire come far funzionare il tutto. Non vi nascondo che la cosa è abbastanza frustrante per un utilizzatore abituato al terminale, ma questi erano i sistemi dei quali si disponeva trenta anni fà! Consiglio vivamente a coloro che non hanno mai praticato la macchina una lettura dell'ottimo help in linea che spiega sia i principi base che gli aspetti meno immediati e più "sfiziosi" dell'emulatore in funzione.

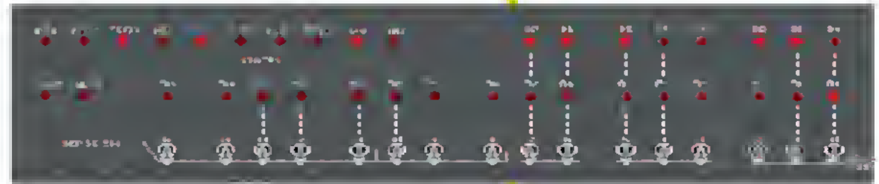
In ogni caso prima di partire conviene individuare gli elementi principali dell'interfaccia. Gli elementi fondamentali sono i led che indicano gli otto bit di una parola di dati (da D0 a D7) e i 16 led delle linee di indirizzo (A0...A15). Questi led si accendono se il corrispondente bit è a uno e si spengono con il bit a valore zero.

Ovviamente essi possono indicare un singolo byte o un indirizzo alla volta. E' possibile comunque esaminare tutta la memoria operando sui bottoni del pannello frontale.

Ad esempio a macchina appena accesa abbiamo tutti i led di indirizzo spenti, cioè stiamo vedendo l'indirizzo zero (0000 0000 0000 0000 in binario) e il valore della cella di RAM a questo indirizzo è (vedi led accesi per i dati) 1011 0010:



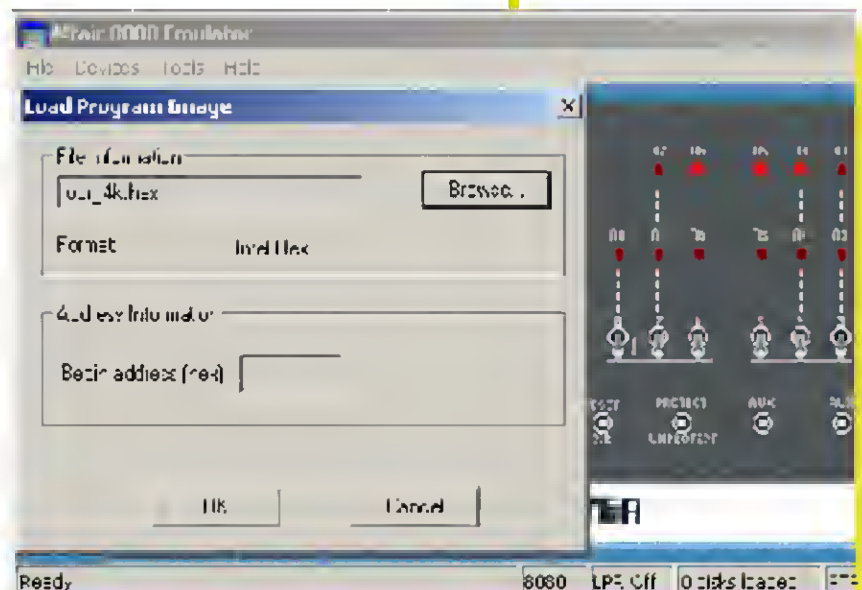
Un click su Exame Next ci fa scoprire cosa contiene l'indirizzo 1 (vedi figura sotto), e così via.

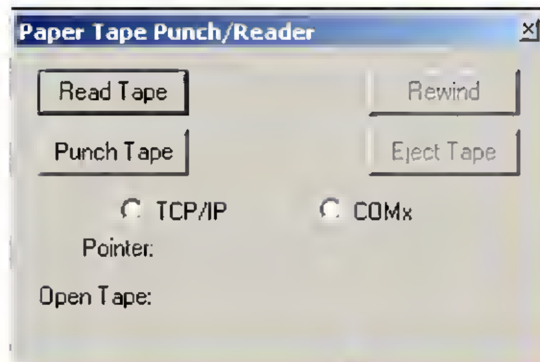


La funzione degli switch è duplice: servono per impostare gli indirizzi e i dati, ma anche per ordinare le funzioni (store, load, etc...) oltre che guidare certe funzionalità di "monitor" del sistema che permettono ad esempio di indirizzare l'output verso una periferica piuttosto che un'altra.

Non è questa la sede per descrivere le operazioni che permettono di usare il computer e per le quali rimandiamo alla prova della macchina. Ci limiteremo ad esplorare le funzionalità dell'emulatore.

Comode funzioni di dump e recupero della memoria, rendono agevole l'utilizzo dell'emulatore.





La maschera, fin troppo spartana, del lettore/perforatore di nastri cartacei. Come si intuisce dai controlli presenti, è anche un sistema per passare dei dati all'emulatore attraverso la comunicazione via rete o via seriale.

Una nota di servizio e di "merito", se vogliamo è la presenza di un debug integrato che, accessibile dal menu Tools è di enorme aiuto per chi volesse cimentarsi con l'utilizzo del sistema via front panel lavorando tutto in binario. Le finestre del debugger permettono infatti l'esame della memoria e dei registri della CPU.

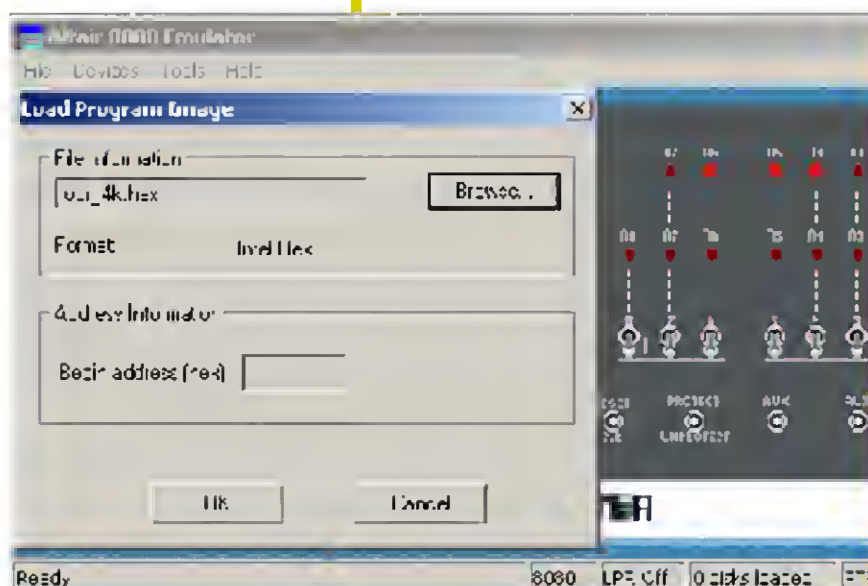
Le periferiche

Torniamo al sistema emulato per occuparci delle periferiche che possiamo collegare allo stesso (in maniera virtuale, si intende).

La meno consueta delle periferiche è senza dubbio il lettore/registratore di nastro di carta, chiamato "Punch/Reader" che, come senza dubbio saprete, consiste nella possibilità di perforare un nastro di carta in maniera da rappresentare un byte per ogni posizione e ovviamente rileggerlo tramite un meccanismo a contatto elettrico o, più recentemente, ottico.

Come resistere alla voglia di provarla, anche se emulata, questa storica periferica? La finestra di gestione dell'unità permette di lavorare su due tipi di nastro virtuali (le immagini binarie a disco di una bobina) che sono il formato TAP, il più diffuso e comune alle cassette magnetiche, e il formato PTP più specifico di questo tipo di supporti. La gestione della periferica non è dissimile da un lettore di cassette audio: si può leggere e riavvolgere il nastro, caricare un nastro vuoto e registrarlo con il contenuto della memoria. peccato che gli autori del programma non abbiano riservato nel disegno di questa periferica, la stessa cura messa invece per il lettore a nastro. La maschera di gestione è semplicissima: per la lettura ad esempio, dopo aver scelto il nome del file (4kbas32m.tap), contenente il basic microsoft da 4 Kb, si dovrà procedere a fornire i necessari comandi direttamente sul pannello di controllo. Un boot-loader minimo che consente di caricare il contenuto del nastro ed eseguirlo è composto da 21 istruzioni che andranno inserite una alla volta dal FP o, molto più comodamente, caricate nella RAM grazie ad un file di salvataggio già predisposto dagli autori (gli emulatori sono proprio comodi in questi casi!). Attraverso la voce di menu "Read image file..." si procede al caricamento (vedi figura a fianco). Se andiamo ad esaminare il loader

Load di un dump di memoria in formato esadecimale.



appena caricato (immagine nella pagina a fianco), scopriamo che la periferica è mappata all'indirizzo 07h, infatti il loop di caricamento include l'istruzione IN 07h.

A questo punto resettiamo la CPU per impostare i valori corretti dei registri, fra i quali importante il PC che deve puntare all'indirizzo zero, da dove iniziano le istruzioni del caricatore. Da notare che il reset non modifica il contenuto della memoria, comportamento che è corretto perché agisce solo sul processore e non innesca alcuna routine di gestione della macchina. Settiamo lo switch A15 e lanciamo il RUN.

Come si accennava gli switch servono sia per impostare indirizzi e dati e sia per guidare la macchina, ad esempio per indirizzare l'I/O.

Il caricamento è immediato ovviamente, se avessimo dovuto leggere veramente un nastro perforato l'operazione avrebbe impiegato qualche minuto (anche per questo amo gli emulatori!).

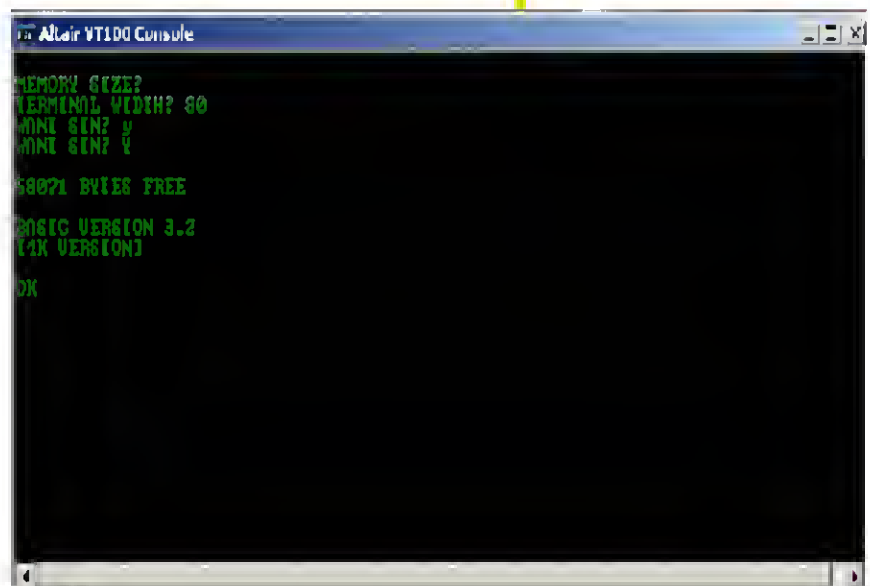
Al termine del caricamento il PC è fermo su una istruzione HLT (halt, cioè di stop del processore), dobbiamo resettare per impostare il PC all'indirizzo zero e dare il RUN. Qui bisogna decidere quale output vogliamo considerare; l'emulatore prevede infatti la possibilità di collegare un terminale vero via seriale o arrivarci con un Telnet. La scelta più comoda è però quella di attivare la console virtuale, una specie di terminale built-in, al momento della configurazione. Per fare in modo che l'output venga mandato al terminale e che l'input per converso sia accettato dalla tastiera, dobbiamo definire la porta di I/O mettendo alto lo switch dell'indirizzo A11. Il Run a questo punto attiva la console virtuale e ci permette di interagire con il basic caricato (vedi figura in questa pagina).

A questo punto ci siamo lanciati nella programmazione (si fa per dire), con il più classico dei cicli FOR e stampa dei primi 20 numeri. Dobbiamo dire che il sistema risulta usabilissimo e anche abbastanza veloce già a 2 MHz, che è il setting predefinito.

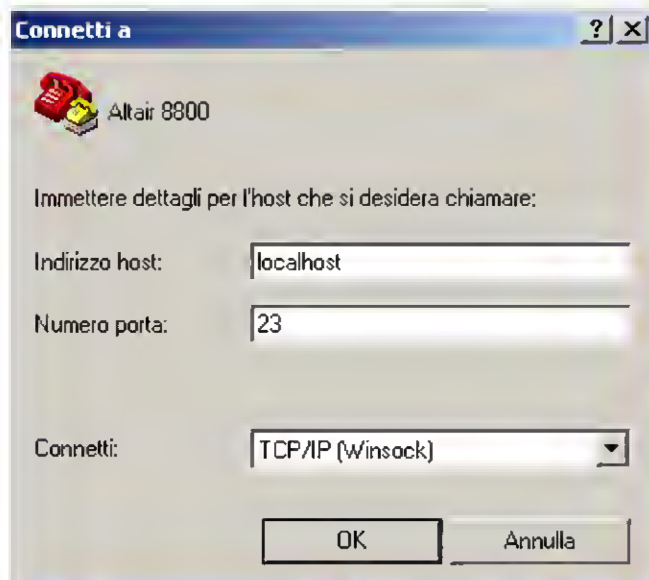
Una modalità di funzionamento che avvicina maggiormente l'uso dell'emulatore alla macchina reale, è la predisposizione di un terminale fisico in collegamento seriale, oppure in subordine la gestione di una console attraverso un programma di emulazione di terminale, come ad esempio Hyperlink che è con-

```
0000 21AE0F     LXI   H,0FAEh
0003 311200     LXI   SP,0012h
0006 DB06       IN    06h
0008 0F         RRC
0009 D8         RC
000A DB07       IN    07h
000C BD         CMP   L
000D C8         RZ
000E 2D         DCR   L
000F 77         MOV   M,A
0010 C0         RNZ
0011 E9         PCHL
0012 03         INX   B
```

Il pezzo di codice che carica dati dal nastro perforato.



Il Basic 4k alla partenza nella console virtuale.

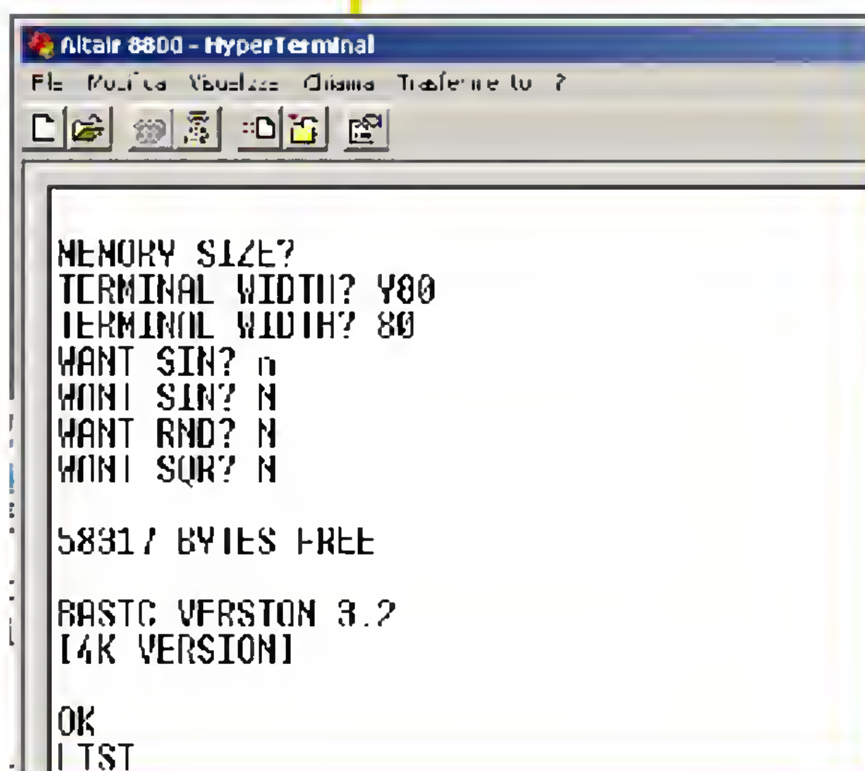


Impostazione dei parametri di HyperLink per collegarsi all'emulatore.

tenuto negli accessori di Windows. In questo caso si deve impostare la console prima di accendere l'Altair sul parametro "Telnet, porta 23" e settare l'hyperlink o qualsiasi altro programma di terminale che supporti questo protocollo all'indirizzo "localhost".

Un aspetto tutt'altro che trascurabile quando si ha a che fare con un emulatore è la possibilità di scambiare dati con un PC. La maniera più semplice, nel caso si voglia trasferire un sorgente BASIC ci viene offerta proprio dall'emulazione di Hyperlink come console (ma anche altri emulatori di terminale hanno analoghe possibilità).

Il BASIC Microsoft caricato nell'emulatore di terminale.



Il sorgente BASIC viene trasferito "come fosse digitato sulla tastiera dell'Altair". Un bel vantaggio, non c'è che dire...

Più famigliare del punch reader è il registratore a cassette. L'emulazione è curata anche dal punto di vista grafico, compreso il contatore di giri che si incrementa durante la lettura del nastro. Ovviamente la periferica si controlla come il punch reader caricando gli stessi formati di nastro. Anche dal punto di vista della velocità sono comparabili le due cose visto che il supporto fisico è un file a disco non ci sono ritardi di alcun genere e la velocità di scorrimento del nastro è del tutto ininfluente.

Ovviamente è possibile emulare anche la presenza di una stampante, utilissima periferica ai tempi dell'Altair e anche dopo. Qui gli autori dell'emulatore non si sono sprecati più di tanto, nel senso che esiste si una immagine del meccanismo di stampa, ma l'output viene appoggiato solo su un file a disco, senza mostrarlo durante la generazione dello stesso.

Floppy e CP/M

Attaccare un controller per floppy disk con una o più unità rappresentava all'epoca dell'Altair originale un lusso riservato a pochi nababbi (circa 1700 dollari a disco). Per noi che emuliamo le cose sono decisamente più facili. Come altri emulatori anche Altair32 emula la

presenza dei floppy con le loro immagini binarie residenti sul sistema host.

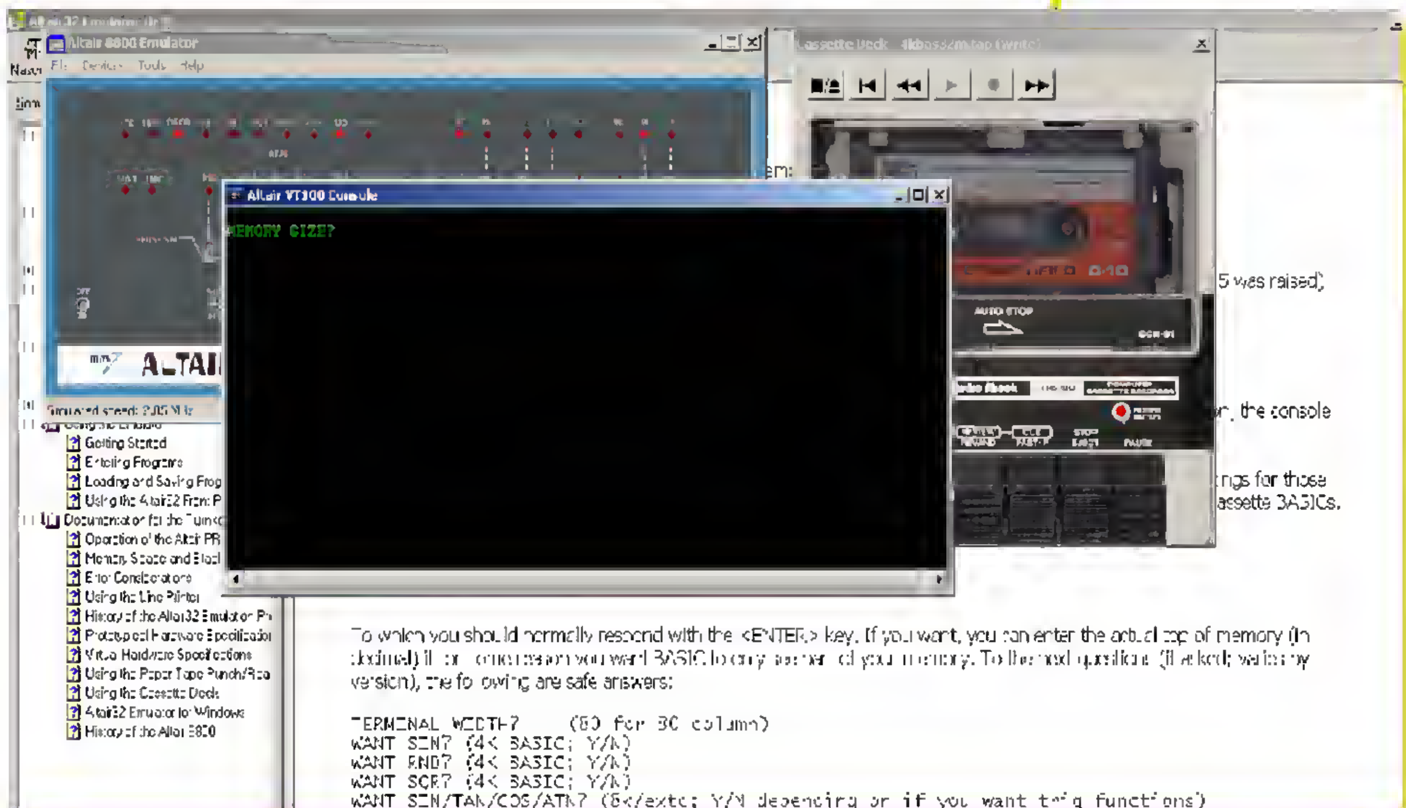
Il primo passo è “attaccare” l’unità al controller attraverso il menù dei device e scegliere l’immagine del floppy opportuna (ad esempio quella per il CP/M 2.2, fornita con l’emulatore); successivamente bisognerà fare boot questa volta usando la routine di caricamento predisposta nella PROM della macchina all’indirizo 0xFF00. Si agisce come di consueto impostando il Program Counter e lo switch A15 prima di dare il run. Il led read/write dell’unità si illumina e se abbiamo abilitato anche i suoni si possono ascoltare i magici ronzii delle parti meccaniche, una vera gioia per le nostre orecchie!

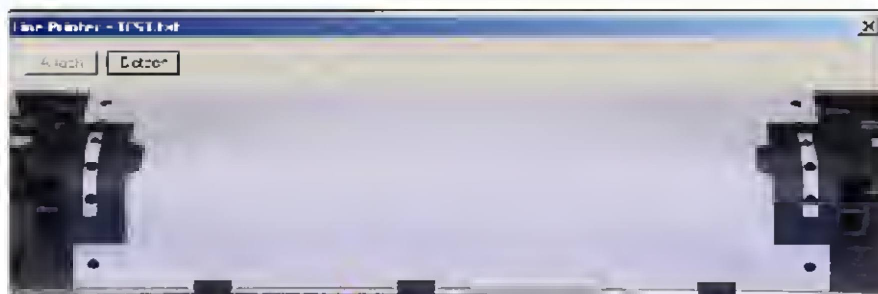
Il CP/M viene fornito nella più classica delle versioni 2.2 e non presenta alcuna particolarità rispetto ad altri sistemi. Un comando DIR ci elenca il contenuto del dischetto “master” che contiene i classici software di utility e l’interprete MBASIC. Con l’interprete della Microsoft ci si può trovare meglio nel tentare di eseguire programmi, anche di una certa complessità , forniti come esempio, come il gioco CHESS che con i suoi 14 Kb di sorgente Basic che vanno ad occupare una buona fetta di memoria.



Il lettore di cassette audio. Notare la presenza del contatore di giri; invece i tasti del registratore non servono nell’emulazione che fa uso della striscia di comandi visibile nella parte alta.

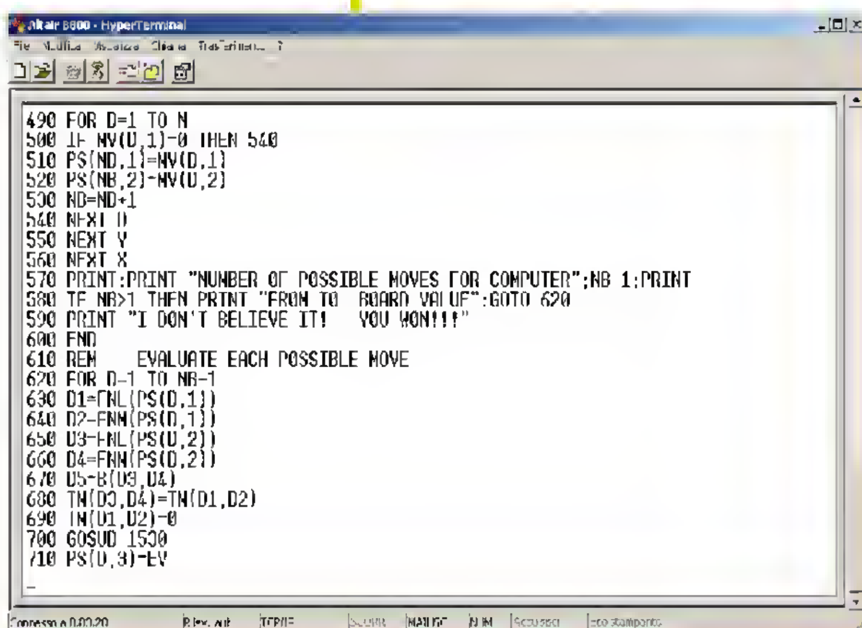
Oltre al floppy l’emulatore ha la possibilità di gestire una unita di





La stampante emulata, una classica a matrice da 80 colonne.

grandi dimensioni, chiamiamola hard disk. Probabilmente una periferica di memorizzazione di questa classe non era disponibile ai tempi dell'uscita del sistema e se lo era non riusciamo ad immaginare il costo, che deve essere stato comunque spropositato. Il solito file a disco consente di fare ciò , mentre una versione "con steroidi" del CP/M lo gestisce come fosse un "floppone" di grande dimensione.



```
490 FOR D=1 TO N
500 IF NV(D,1)=0 THEN 540
510 PS(NB,1)=NV(D,1)
520 PS(NB,2)=NV(D,2)
530 NB=NB+1
540 NEXT D
550 NEXT Y
560 NEXT X
570 PRINT:PRINT "NUMBER OF POSSIBLE MOVES FOR COMPUTER";NB-1:PRINT
580 IF NB>1 THEN PRINT "FROM TO   BOARD VALUE":GOTO 620
590 PRINT "I DON'T BELIEVE IT!   YOU WON!!!"
600 END
610 REM    EVALUATE EACH POSSIBLE MOVE
620 FOR D=1 TO NB-1
630 D1=FNL(PS(D,1))
640 D2=FNM(PS(D,1))
650 D3=FNL(PS(D,2))
660 D4=FNM(PS(D,2))
670 D5=B(D3,D4)
680 TN(D3,D4)=TN(D1,D2)
690 TN(D1,D2)=0
700 GOSUB 1500
710 PS(D,3)=EV
```

Il programma Chess. Sopra una parte del listato e sotto l'output.

```
RUN

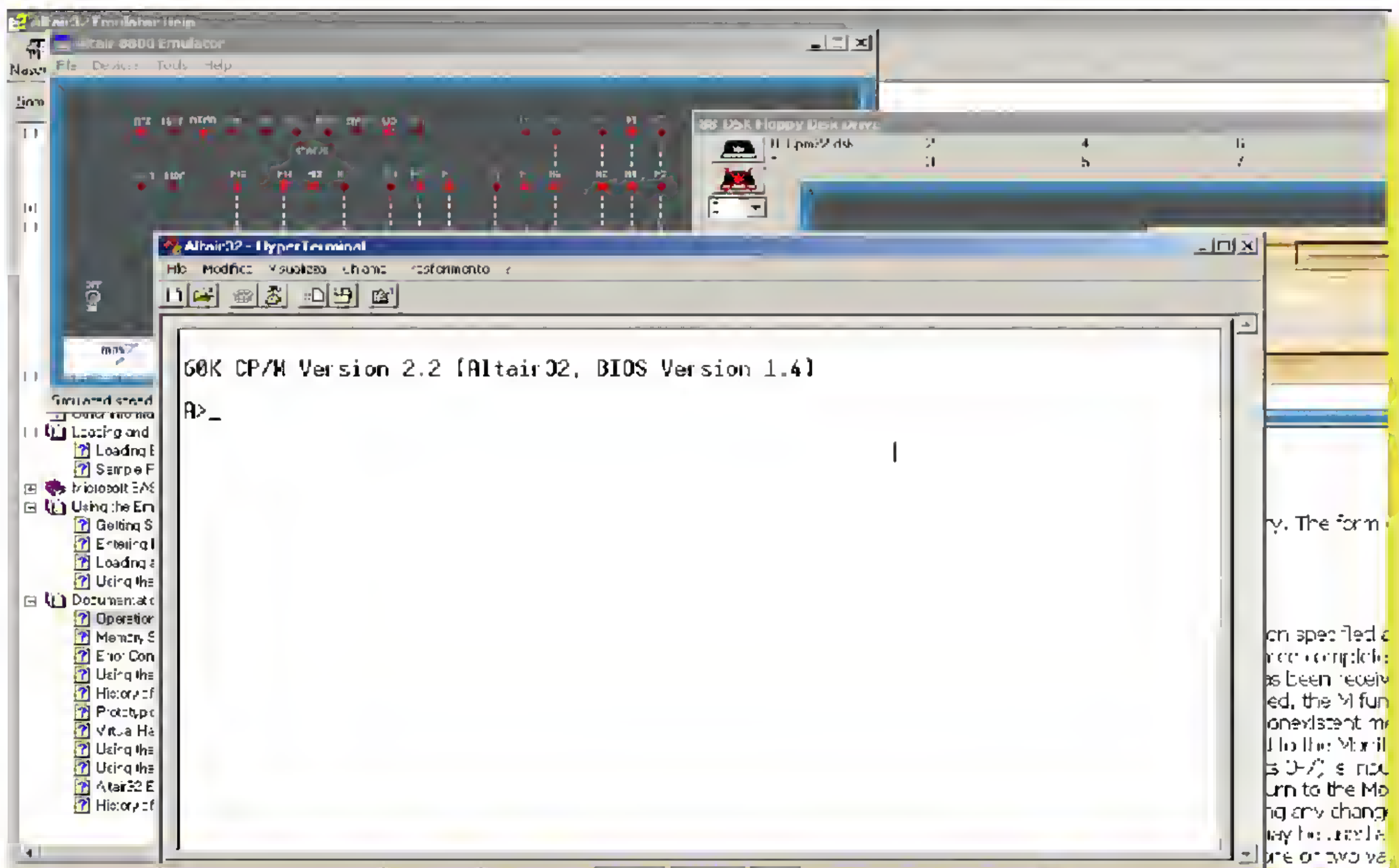

          **** CHESS ****
DO YOU WANT TO BE WHITE OR BLACK? W
 4  2  3  5  6  3  2  4
 1  1  1  1  1  1  1  1
 0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0
-1 -1 -1 -1 -1 -1 -1 -1
-4 -2 -3 -5 -6 -3 -2 -4
YOUR MOVE?
```

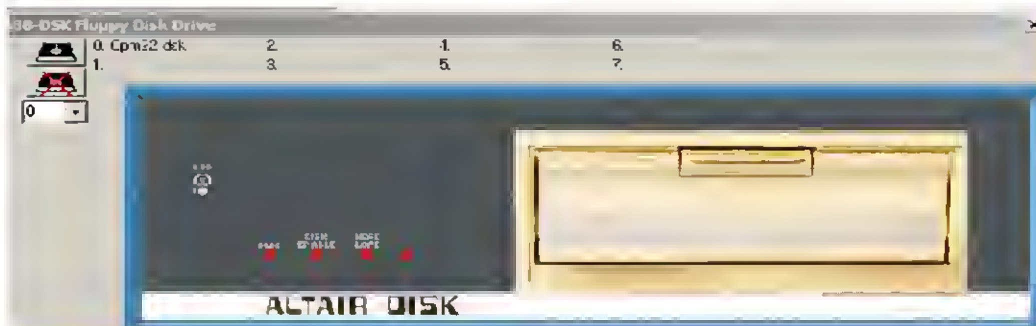
Conclusione

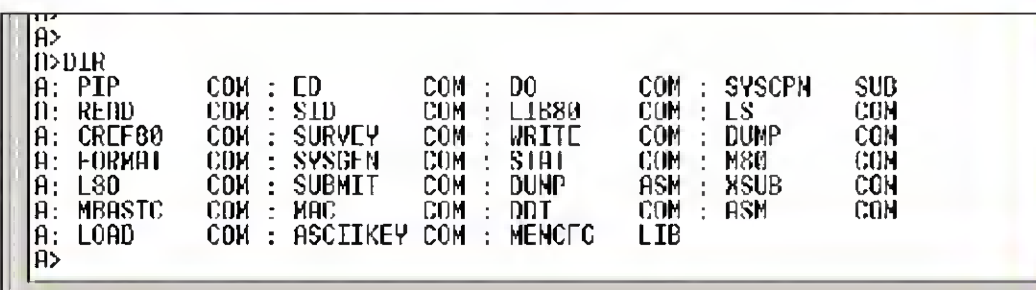
Altair32 è l'emulatore che mancava! Intendiamoci, tutte le iniziative di emulazione sono preziose per conservare la memoria dei tempi che furono; ma già sapete come la penso su questo punto... Nel caso dell'Altair sarebbe comunque molto difficile mettere le mani su una console vera ed è anche per questo motivo che dobbiamo essere grati all'autore e a quanti hanno collaborato alla messa a punto di questo prodotto.

L'atmosfera che si percepisce usando questo programma è vicina alle senzazioni che potevano provare i reali utilizzatori. Immagino una stanza illuminata solo dal monitor e dalle luci intermittenti del pannello frontale e un giovane studente alle prese con questo nuovo entusiasmante mondo... vera poesia.

[L2]







Tre immagini di utilizzo dell'emulatore con l'unità floppy. Sopra il prompt del CP/M appena caricato. Al centro la finestra di gestione dell'unità ed infine la lista del contenuto del Master Disk, non dissimile da ogni altro CP/M 2.2 disponibile per altre piattaforme.

Apple Club

La mela come paradigma della programmazione

Tutti i linguaggi dell'Apple (6)

Microsoft BASCOM

Pensavamo di presentare il compilatore per BASIC-80 nell'articolo del fascicolo precedente, ma poi lo spazio, sempre tiranno, non lo ha consentito. A dire il vero c'è un'altro motivo: sarebbe stato un peccato esaurire in poche righe un oggetto di tale portata.

Si tratta di un compilatore BASIC in particolare adatto a trasformare in codice oggetto i sorgenti scritti e testati con l'interprete BASIC-80, sempre sotto CP/M.

Come era logico aspettarsi il compilatore non è in grado di tradurre il 100% degli statement disponibili. Infatti il BASIC oltre che essere un linguaggio di programmazione, negli home e personal funge anche da ambiente di comando. Non disponendo di un sistema operativo infatti, quel minimo di interazione con le periferiche e in particolare con i registratori magnetici, viene portato a termine proprio dall'interprete.

Le istruzioni proibite sono elencate nel box 1; si tratta come vedete di direttive e comandi che evidentemente non hanno senso in un programma compilato.

Nel box n. 2 sono riportate invece quelle istruzioni che pur facendo parte del bagaglio del compilatore hanno però un comportamento diverso rispetto all'ambiente interpretato e se ne dovrà pertanto tenere conto quando sono utilizzate.

La compilazione è guidata dalla parte di switch che permette di includere o escludere certe librerie.

Ecco la lista di questi switch:

/4 = riconosce le convenzioni del BASIC Microsoft 4.51 e seguenti.

/T = riconosce le convenzioni dell'interprete MBASIC dalla release 4.51.

Comandi di compilazione.

Per invocare la compilazione:

```
BASCOM command line
```

dove "command line" è una stringa di parametri così composta:

```
[dev:][obj file][,[dev:][lst file]]=[dev:]source file[/switch ...]
```

Al solito le parentesi quadre raggruppano parametri opzionali. L'unico necessario è il nome del file da compilare, in formato ASCII e salvato su una unità raggiungibile e nota al CP/M (normalmente sarà l'unità B: se in A: abbiamo il dischetto del compilatore.

Mi accorco di essermi dimenticato di precisare che di interpreti BASIC per CP/M la Microsoft in realtà ne ha fatto uscire due: il BASIC-80 e l'MBASIC, poi trasportato anche in DOS.

/C = non tiene conto delle convenzioni BASIC-80.

/E = usare se nel sorgente viene usato il trapping degli errori con la sequenza di statement: ON ERROR GOTO e RESUME <line number>.

/Z = usa le istruzioni dello Z80 (di default utilizza solo quelle dell'8080).

/N = non produce la lista del disassemblato.

/D = genera il codice per il debug e per il trattamento degli errori a run-time.

/S = scrive in output, durante la compilazione, le stringe alfanumeriche presenti nel sorgente, se sono lunghe più di quattro caratteri.

Produrre un programma eseguibile era operazione un attimino più complicata rispetto a quanto avviene con i moderni compilatori. E' comunque vero che in generale la sequenza di operazioni è rimasta più o meno la stessa, semmai oggigiorno certe operazioni sono compiute in automatico in maniera, diciamo così trasparente, al programmatore.

Vediamo in dettaglio quello che occorre per produrre un eseguibile tramite BASCOM sotto CP/M.

***Preparare il sorgente**. La prima operazione è quella di creare il sorgente e salvarlo su disco in formato ASCII. Questo può essere prodotto inserendo gli statement nel normale interprete (che deve avere una sintassi compatibile con il compilatore), oppure creando un file sorgente ASCII con qualsiasi editor o word processor in grado di farlo. Operando con l'interprete si ha ovviamente il vantaggio di debuggare il sorgente mentre si edita, ma le funzioni di modifica del testo insite in un più adeguato strumento potrebbero rendere di molto più agevole la stesura di procedure di un certo impegno.*

Le regole per dotare di un nome il sorgente, seguono le convenzioni del CP/M (nomi di otto caratteri) con estensione BAS.

***Syntax check**. Per avere un controllo sulla bontà del sorgente, ad esempio per scoprire errori di sintassi o istruzioni non consentite, si può procedere ad una "finta" compilazione con il comando:*

BASCOM =myprog

dove "myprog" è ovviamente il nome del sorgen-

AUTO
CLOAD
CSAVE
CONT
DELETE
EDIT
ERASE
LIST
LLIST
LOAD
MERGE
NEW
RENUM
SAVE

Box 1.
Lista delle parole chiave non riconosciute dal compilatore.

Box 2.
Lista delle istruzioni che si comportano diversamente nei due ambienti.

CALL
CHAIN
CLEAR
COMMON
COS
DEFINT
DEFSNG
DEFDBL
DEFSTR
DIM
END
ERASE
EXP
FOR/NEXT
LOG
ON ERROR GOTO
RESUME n
REM
RUN
SIN
STOP
SQR
TAN
TROFF
TRON
USRn
WHILE/WEND
%INCLUDE

te salvato sul disco. L'estensione è opzionale se si è usato ".BAS" come consigliato.

Compilazione. La compilazione vera e propria produce due file su disco che hanno lo stesso nome del sorgente ed estensioni rispettivamente ".REL" per codice oggetto rilocabile e ".LST" per il file di listing.

A parte il file di listing che potrebbe anche essere soppresso se non ci interessa curiosare sul lavoro fatto dal nostro compilatore, il file oggetto contiene le istruzioni pronte per produrre il codice macchina, cosa che avviene invocando il programma che effettua il linker.

Linking. Dal codice oggetto rilocabile si ottiene un eseguibile (formato COM) con il comando:

```
L80 myprog,myprog/N/E
```

questi scrive su disco un file eseguibile che si chiamerà "myprog. COM". Il nome dello stesso sarà sufficiente per mandare in esecuzione il programma stesso.

Tutti questi comandi hanno la loro sintassi e i loro flag da usare secondo le circostanze. Noi abbiamo semplificato all'osso la sessione di compilazione, operando secondo il default. La cosa che succederà nella maggior parte dei casi sarà che quantomeno il sorgente e il compilato dovranno risiedere su un dischetto diverso rispetto a dove si trovano i comandi. In questo caso l'indicazione "B:" prima del nome del file fornisce la specifica necessaria a coinvolgere la seconda unità magnetica collegata al computer e gestita dal CP/M.

In particolare il Linker/Loader carica in memoria un file in formato REL, e lo "accomoda" in memoria in modo che sia eseguibile a partire dall'indirizzo 100h.

Lo switch /N dice a L80 di salvare una immagine della memoria sul disco con espensione .COM.

Lo switch /E semplicemente comanda al linker di uscire al CP/M una volta terminato il lavoro.

Una cosa da ricordare è che al momento del link devono essere presenti le funzioni di libreria utilizzate all'interno del codice. Ad esempio se il sorgente effettua una chiamata di funazione ad una qualche funzione presente in una libreria FORTRAN compatibile, la libreria stessa deve essere indicata al momento del link:

```
L80 myprog, forlib/S, myprog/N/E
```

La differenza di esecuzione fra interprete e compilatore viene dichiarata variabile dal 10% fino al 30%, quando si operi una attenta organizzazione del sorgente mettendo in attop tutti quegli accorgimentin che permettono al compilatore di ottimizzare il codice.

Un test con BASCOM

Spinti da quella che potremmo definire "una sana curiosità", volevamo toccare con mano l'incremento di velocità rispetto all'interprete. Abbiamo così rispolverato il buon vecchio bubble sort e messo alla prova.

Nel box a fianco riportiamo il sorgente. Si tratta di ordinare un vettore di dieci elementi iterando l'operazione per 30 volte. Vogliamo dirvi subito che le prove non sono state rigorosissime e i tempi presi a mano con approssimazione.

Abbiamo chiamato questo sorgente "BUBBLE.BAS" e salvato in formato ASCII.

Il Basic80 Microsoft, versione 4.5, esegue il lavoro in circa 33 secondi sul nostro sistema di prova (un Apple II).

Compilato e linkato per produrre l'eseguibile BUBBLE.COM (la compilazione più il linking è durata circa 5 minuti), abbiamo lanciato questo eseguibile più volte per fare una media delle esecuzioni.

Ebbene il risultato è stato di appena sei secondi (contro i 33 dell'interpretato); davvero un ottimo risultato!

```
10 DEFINT I-S
20 DIM A(10)
25 PRINT "START"
30 FOR I = 1 TO 30
40 A(0)=9:A(1)=4:A(2)=3:A(3)=8:A(4)=7:A(5)=1:A(6)=2:A(7)=4:A(8)=6:A(9)=3
50 S = 1
55 REM
60 WHILE S > 0
65 S = 0
80 FOR J = 1 TO 9
90 IF ( A(J-1) > A(J) ) THEN GOSUB 200
100 NEXT J
110 WEND
120 NEXT I
123 GOSUB 130
124 PRINT "END"
125 END
130 FOR K=0 TO 9
131 PRINT A(K),
132 NEXT K
135 RETURN
200 H = A(J-1)
210 A(J-1) = A(J)
220 A(J) = H
225 S = 1
230 RETURN
```

Conclusioni

Decisamente niente male questo prodotto: discretamente veloce nel compilare e nel risultato ottenibile, è anche abbastanza flessibile da non dover stare particolarmente attenti nella creazione del codice per l'interprete, passaggio obbligato della messa a punto, come abbiamo detto nell'articolo.

[Sm]

Sopra: il listato del programma per il bubble sort;
sotto: una parte del file di listing prodotto dalla compilazione.

```
BASCOM 5.2 - Copyright 1979, 80 (C) by MICROSOFT - 20202 Bytes Free
 0014 0007       10 DEFINT I-S
        ** 0014'L00010:
 0014 0007       20 DIM A(10)
        ** 0014'L00020:
 0014 0033       25 PRINT "START"
        ** 0014'L00025: CALL    $PR0A
        ** 0017'        LXI     H,<const>
        ** 001A'        CALL    $PV2D
 001D 0033       30 FOR I = 1 TO 30
        ** 001D'L00030: LXI     H,0001
        ** 0020'        JMP     I00000
        ** 0023'I00001:
 0023 0033       40 A(0)=9:A(1)=4:A(2)=3:A(3)=8:A(4)=7:A(5)=1:A(6)=2:A(7)=4:A(8)=
6:A(9)=3
        ** 0023'L00040: CALL    $FASA
        ** 0026'        DW      A!
        ** 0028'        DW      <const>
        ** 002A'        CALL    $FASA
        ** 002D'        DW      A!+0004
        ** 002F'        DW      <const>
        ** 0031'        CALL    $FASA
        ** 0034'        DW      A!+0008
        ** 0036'        DW
```

Retro Riviste

La rassegna dell'editoria specializzata dai primi anni '80 ad oggi

Scheda

Titolo: MICRO

Sottotitolo: The 6502 Journal

Lingua: Inglese

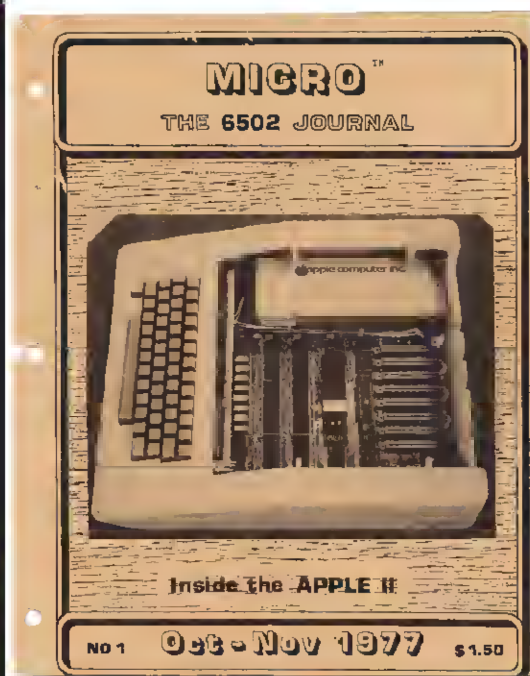
Prezzo: 1,5 $

Pagine: 34-72

Primo numero: Ottobre 1977

Ultimo numero: Ottobre 1984

MICRO



Qui siamo proprio agli albori della storia dei microcalcolatori e come è noto questa industria è stata gemmata direttamente dall'elettronica.

Ovvio aspettarsi l'uscita di riviste specializzate nell'elettronica digitale e nei microporcessori in particolare.

Il titolo che esaminiamo oggi è dedicato espressamente ad uno dei micro processori che hanno avuto maggiore fortuna nel settore: quel 6502 che ha equipaggiato decine di home per tutto il decennio '80 del secolo scorso.

Suona un po' strano parlare del "secolo scorso", evidentemente non ci siamo abituati ancora a considerare il 900 come un'epoca passata! Mi viene anche spontanea la considerazione di come il 900 sia iniziato e finito con due rivoluzioni scientifiche/tecnologiche: la teoria della relatività Speciale di Einstein (nel 1905) e la rivoluzione informatica di fine secolo.

Si capisce chiaramente fin dal titolo quale sarà la logica redazionale della rivista MICRO e cioè parlare dei progetti che hanno come cuore pulsante il chip della Rockwell (e di altri partner).

La copertina del primo numero è dedicata al "principe" dei calcolatori personali che fanno uso di questa CPU: sua maestà l'Apple][. Copertina e "strillo" che annuncia "Inside the Apple II" devono aver avuto una discreta presa sugli appassionati che immaginiamo impegnati nel discutere attorno al primato di questa o quella CPU. Si sa infatti che il mondo si divide sempre in fazioni, possibilmente due, ognuna delle quali disposta a giurare carte false pur di vedere la propria convinzione assunta a verità sacrosanta.

Mi obbliga una doverosa precisa-

zione ricordando che sarebbe stato difficile reperire in Italia simili prelibatezze, visto che l'importazione della macchina Apple deve essere avvenuta ufficialmente non prima del 1980. Questo lo sò per certo visto che nel 1980 ho avuto occasione di conoscere personalmente (e casualmente) uno dei soci della Harden che stava pianificando proprio l'importazione ufficiale degli Apple serie 2. Queste persone sapevano esattamente dove volevano arrivare: ad un mercato di massa come quello statunitense. Solo che non sapevano come arrivarci e se in Italia avrebbe mai funzionato!

I micro nel 1977 erano pochini; giusto l'Apple, come si diceva, il KIM-1 e pochi altri. Questa "povertà" potrebbe apparire una sfortuna oggi, ma in qualche modo era invece una cosa opportuna: permetteva di diedicarsi in maniera approfondita ad un tema, senza paura di vedersi sfuggire la tecnologia davanti al naso.

Il primo numero della rivista è dell'ottobre 1977, l'ultimo (almeno del quale io abbia notizia) è esattamente di sette anni dopo: ottobre 1984. Credo che la publicazione abbia continuato in forma autonoma o magari "embedded" in qualche altra pubblicazione, dal momento che il 1984 erano ancora in auge i sistemi con il 6502.

Dalla ricchezza di temi elettronici dei primi numeri, si è passati pian pianino verso il software come focus primario della pubblicazione.

E' una transizione abbastanza comune, se non universale, delle pubblicazioni "della prima ora", quella di evolversi verso il lato "soft" dei sistemi. Questo a mano a mano che gli utilizzatori non tecnici prendono il sopravvento sugli "smanettatori" armati di saldatore.

Così anche l'assoluta devozione al 6502 viene pian pianino affiancata dalla desamina degli altri micro e non solo per mero confronto prestazionale. Ad esempio il 6800 Motorola, in un certo senso considerato un fratello maggiore del 6502, è preso in seria considerazione. Non mancano peraltro articoli sullo Z80 che non poteva certo essere ignorato, vista la sua diffusione almeno pari, se non superiore rispetto al concorrente.

In occasione nel fascicolo 37 il sottotitolo diventa "The 6502/6809 Journal" per abbandonare definitivamente le sigle dei processori e diventare "Advancing Computer Knowledge".

Una rivista molto interessante per tutti coloro che amano il retro computing.

[Sn]



A fronte la copertina del primo numero e qui sopra quella del numero 76, forse l'ultimo uscito.

TAMC

Teoria e Applicazioni delle Macchine Calcolatrici: la matematica e l'informatica, le formule e gli algoritmi, la completezza e la computabilità, le strutture dati e tutto quello che sta alla base dell'informatica.

Algoritmi di SORT (parte 1)

Vogliamo cominciare con questo articolo una serie di pezzi su uno dei "tormentoni" dell'informatica: l'ordinamento. Praticamente non c'è testo o corso che insegni le basi delle teorie informatiche o semplicemente i principi della programmazione dei calcolatori, che non ne tratti più o meno approfonditamente.

Come mai tanto interesse per questo argomento? Verrebbe da chiedersi... Si tratta sì di un aspetto importante della programmazione, un problema che si incontra frequentemente nella vita "programmatoria" di tutti i giorni, ma c'è dell'altro. Non per nulla i francesi chiamano il calcolatore "Ordinatour", evidenziando in maniera inequivocabile fin dal nome che si tratta di una macchina in grado di ordinare le cose. L'ordinamento è in generale un argomento affascinante e che si presta ad illustrare molteplici aspetti della programmazione e di quant'altro è legato all'idea del calcolo automatico.

Vi invitiamo pertanto a seguirci in questo piccolo ma affascinante e per quanto ci è possibile completo, viaggio nel mondo dell'ordinamento.

Principi Generali

Ricordiamo brevemente per ragioni di completezza di trattazione, che qualsiasi problema di ordinamento è riconducibile ad un ordinamento di un vettore di interi. Se indichiamo con N la dimensione del vettore da ordinare, cioè il numero di elementi in esso contenuti, l'operazione verrà compiuta da un algoritmo che in generale prevede X passi, con X funzione di N; $X = f(N)$.

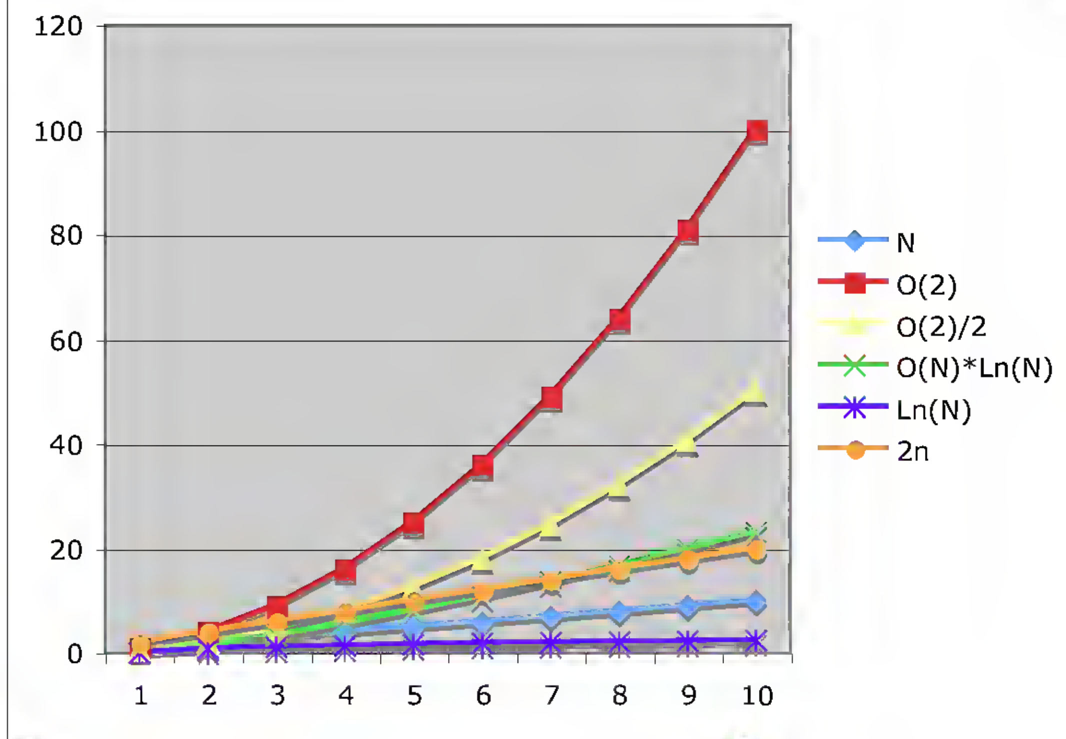
In relazione all'algoritmo scelto la complessità del problema avrà un certo ordine; ad esempio $O(\ln(N)*e^N)$ indica un algoritmo che al crescere di N impiega un tempo proporzionale alla funzione $\ln(N)*e^N$.

Per la cronaca un algoritmo che avesse questo tipo di complessità sarebbe particolarmente oneroso in termini di tempo di calcolo, tale da risultare improponibile nol caso si debbano trattare problemi con N molto elevato.

Già un andamento lineare al crescere di N pone seri problemi di effettiva computabilità. L'ideale sono gli algoritmi (purtroppo non sempre possibili) che hanno un andamen-

to "descrescente" rispetto alla retta lineare. Ad esempio un algoritmo con complessità log(N) è buono, come anche $O(N^{1/2})$.

La ricerca di un algoritmo efficiente, cioè che consumi meno tempo possibile per essere portato a termine, è uno degli aspetti principe nella ricerca algoritmica. Altri principi sono la correttezza e l'ottimizzazione delle risorse diverse dal tempo.



Un esempio grafico per comprendere il differente comportamento degli algoritmi al crescere di N.

Selection Sort

Come dice il nome questo algoritmo fa suo il principio di selezione dell'elemento minore scelto fra quelli da ordinare. Una volta individuato tale elemento esso va a posizionarsi nella posizione che gli compete all'interno di un vettore ordinato.

Questo algoritmo è l'esatta trasposizione di quello che faremmo dovendoci trovare a scegliere degli oggetti da un insieme per metterli nell'ordine di grandezza in un'altro contenitore.

E' semplicissimo illustrarne il funzionamento se si assume che il vettore ordinato sia un'area diversa da quello da ordinare. In pratica abbiamo bisogno di un vettore risultato della stessa dimensione di quello originale.

Sia [3, 2, 4, 1, 5] il vettore da ordinare e sia inizialmente vuoto il vettore ordinato:

[0, 0, 0, 0, 0]

prima iterazione:

viene scelto l'elemento che vale 1 e spostato nella prima posizione del vettore ordinato:

[1, 0, 0, 0, 0]

L'elemento 1 non fa più parte del vettore da ordinare che ora è:

[3, 2, 4, 99, 5]

Si vede che al posto dell'elemento estratto dal vettore di partenza è stato posto un numero "grande" che si assume faccia la funzione di "posto vuoto".

Ora vien scelto il 2:

[1, 2, 0, 0, 0]

e così via fino ad aver esaurito il vettore di partenza.

Ovviamente si considerare il problema complementare, cioè andare a cercare gli elementi più grandi e costruire il vettore ordinato partendo dal fondo; le due cose sono del tutto equivalenti.

Già la descrizione verbale di quello che avviene ci indica che per un vettore di N elementi saranno necessarie N iterazioni, ognuna delle quali andrà a cercare l'elemento minimo nel vettore, quindi effettuando N confronti.

Dovendo passare alla codifica di un programma per calcolatore faremo uso di una notazione pseudo-pascal che è abbastanza facile da comprendere e molto usata nella descrizione degli algoritmi.

Possibile implementazione:

```
minimo := 99;
for j := 1 to N do begin
   indice := 0;
   for i:=1 to N do begin
     if V[i] <  minimo then
       minimo := V[i];
     indice := i;
   end;
   K[j] := V[indice];
   V[indice] := 99;
   end;
end;
```

E' facile intuire che la complessità dell'algoritmo proposto è O(2), cioè cresce con il quadrato del valore N, numero di elementi che costituiscono il vettore da ordinare.

Si ricorda che la valutazione della complessità viene fatta considerando il termine con l'esponente più alto in una ipotetica rappresentazione polinomiale dell'andamento dell'algoritmo.

Insertion Sort

Fare l'ordinamento utilizzando una area dati pari alla dimensione del vettore da ordinare è ovviamente un compromesso che semplifica l'algoritmo ma che potrebbe essere una inaccettabile soluzione rispetto allo spazio occupato in memoria dal programma.

In effetti si può anche effettuare il lavoro direttamente sul vettore originale provvedendo a scambiare gli elementi all'interno di un doppio loop:

```
for i := 1 to N do begin
   for j := i+1 to N do begin
   if V[j] < V[i] then
      scambia(V[i], V[j]);
   end;
end;
```

Il vettore viene ordinato "sul posto" e viene diviso in due dall'indice i che lo percorre nel primo loop. Gli elementi prima di i appartengono al vettore ordinato, quelli dopo sono ancora da ordinare e costituiscono il "serbatorio" da dove si pesca l'elemento minore da mettere nella posizione i-esima.

L'algoritmo proposto prende il nome di "Insertion Sort" e, se vogliamo trovare un paragone pratico, è quello che applichiamo quando ordiniamo le carte da gioco che abbiamo in mano: ne preleviamo una e la "inseriamo" al posto giusto per avere alla fine una "mano"

di carte nell'ordine che preferiamo.

L'applicazione dell'algoritmo sul vettore di prova che abbiamo considerato:

[3, 2, 4, 1, 5]

Produce la seguente sequenza di risultati relativi alla corrispondente iterazione dell'indice i:

[1, 2, 4, 3, 5]

[1, 2, 4, 3, 5]

[1, 2, 3, 4, 5]

Quello che abbiamo illustrato è sicuramente il più semplice codice da scrivere per ordinare una serie di elementi "on place", cioè senza aree di memoria di appoggio.

Questo tipo di algoritmi sono definiti anche "statici" o "rigidi" per il fatto che il numero di passi è predefinito qualsiasi sia il vettore di input. La complessità è O(2), come è facile dedurre dall'analisi dei termini di valutazione per un vettore di N elementi:

(N - 1)*((N – 1) + (N – 2) + (1))

N - 1 passi del loop principale ognuno dei quali prevede un numero di confronti decrescente

= N^2 + 2*N + N – 2 = N^2

Considerando l'esponente più elevato, appunto il 2, si determina l'indice di complessità del tutto.

Nella realtà le cose possono rivelarsi diverse, per effetto del fatto che nella media il numero di confronti, assegnazioni, loop, etc... possono essere inferiori rispetto a quanto detta la teoria.

Per valutare l'efficacia di quanto implementato andranno pianificate delle sessioni di test che effettuano numerosi tentativi su insiemi diversi di vettori, in modo da ottenere dalla media dei risultati una indicazione affidabile. E' utile inoltre osservare come questi test andrebbero svolti in un ambiente mono-task, come ad esempio il buon vecchio DOS del PC o qualunque altro sistema Basic-like tipico dei sistemi home di prima generazione. Le moderne tecniche di threading sui sistemi multitasking, come ad esempio Windows, creano tante e tali condizioni di variabilità al contorno da non potersi ritenere affidabili in assoluto.

Conclusione

Abbiamo cominciato a parlare degli algoritmi di ordinamento partendo da quelli più elementari. Selection Sort e Insertion Sort non permettono di andare molto lontano, anche se si rivelano più efficienti di altri quando i numeri sono piccoli, ed è per questo che la gente ha trovato altri modi "più strani" per ordinare le cose con l'aiuto di una macchina.

Nelle prossime puntate di questa serie ne vedremo qualcuno.

[Sm]

Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Novembre 1983



Radio Elettronica & Computer
Anno 12, numero 11, Novembre 1983, Lire 2500

Ad un certo punto dell'era della microelettronica, le riviste che fino a quel momento si erano contese i lettori appassionati dei misteri dell'elettrone e delle onde elettromagnetiche a suon di non so che ingegnosi progetti, si arrabbattarono a convertirsi in riviste di supporto per quella che appariva la nuova passione dilagante e, ancora più interessante per gli editori, nella quale sembrava che la gente avesse genio di spendere. Primariamente ci si accontentò, quasi di malavoglia, a considerare quei pazzi di informatici come buoni per vendere qualche numero in più della rivista o, cercando di tener fede alla propria indole, ingegnandosi ad introdurre il computer, o meglio il microprocessore, quale componente di supporto a progetti che avessero comunque come cuore l'elettronica analogica in bassa o alta frequenza, quale che sia.

Ma dopo poco e specificatamente non tanto per il numero, davvero irrisorio, di coloro che si erano cimentati nella costruzione di un kit a microprocessore, ma più significativamente per l'interessante numerosità di coloro che avevano acquistato o si erano fatti regalare, un Commodore 64 o uno Spectrum, gli argomenti informatici diventarono predominanti. Da una parte ingozzare le pagine della rivista di articoli o meglio di listati, era punto più economico rispetto a mantenere progetti elettronici e dall'altra è legge di mercato il dare al cliente quello che chiede, per ricavarne il massimo profitto... finché dura.

Così qualche testata cambiò nome, qualche altra gemmò un fascicoletto apposito o una più impegnativa iniziativa. Le più arricchirono il titolo per farvi comparire a forza quella parola "Computer" che cosÏ ben si prestava a riassumere tutta l'idea di innovazione necessaria in quel frangente. Una di queste testate è stata Radio Elettronica che, manco a dirlo, diventa di botto "Radio Elettronica &

Computer".

Il taglio editoriale è spartano: niente copertine patinate o carta rasata. Piuttosto una ruvida grammatura già tendente al grigio che si intuisce diventerà gialla nel giro di pochi anni. Il colore è riservato alle copertine, appena appena più ricche in materiale, tutto il resto in bianco/nero o per gli schemi uno sfondo aragosta, evidentemente poco costoso.

La qualità dei progetti elettronici proposti, per quanto di mia conoscenza, appare poco approfondita; evidentemente la rivista non punta molto in alto rispetto alla preparazione dei propri lettori e nella stessa direzione sembra portarsi nel comparto informatica. Come forse saprete o ricorderete un tempo l'informatico hobbista passava il proprio tempo a digitare listati proposti da altri. Non si può dire che tale attività sia priva di valenza didattica, ma per la maggior parte si tratta di noia pura, con magra soddisfazione al momento del RUN, ammesso che il programma abbia voglia di girare senza quei tragici "Syntax Error" che sono il vero incubo dell'informatico di primo pelo.

Così le riviste riempivano pagine e pagine di questi listati più o meno leggibili, offerti alla decifrazione dei volenterosi. Per una qualche idea di accuratezza, davvero mal interpretata, o forse più per spicciare la cosa in poco tempo, questi listati erano trasferiti sulla rivista direttamente dall'output su stampante collegata al relativo home.

Ora, se questo potrebbe apparire come indice di qualità, dato che cos' facendo il lettore poteva essere certo che il programma avrebbe poi girato come doveva, giacché "listato non mente", pur tuttavia la qualità delle periferiche di stampa del tempo era quella che era. Quando andava bene e il nastro inchiostrato era nuovo, il listato poteva anche decifrarsi discretamente ma al consumo di quello o peggio per l'adozione di tecnologie di stampa alquanto esoteriche (a trasferimento di carica elettrostatica), ne risultava un pastrocchio indecifrabile.

Sulla qualità dei programmi c'è poi poco sa dire: ci si accontentava, ecco. Come vedete nella pagina riportata come esempio qui sotto, non è che un programma che allena all'utilizzo delle tabelline sia il massimo della vita, ma ripeto: ci si accontentava.

C'è poi da considerare che la stessa necessità di digitarsi il sorgente "a manina", non lasciava spazio a chissà quali voli di fantasia sull'impegno dei programmi.

Gli home considerati sono quelli del momento: Ap-

ple II, Commodore 64 e Spectrum. Sono già spariti altri protagonisti come il Texas Instruments 99/4A e il Vic 20. Lo ZX80 ormai non lo cita più nessuno mentre qualche rimansuglio del suo successore ZX81 rimane, ma si capisce che non ne avrà per molto.

Non è però tutto da buttare e mi riferisco in particolare ad un ingegnoso controller realizzato per l'Apple][che tramite pochi componenti e due relè, permette di controllare il motore di due registratori a cassette. L'Apple][ha uno zoccolo sulla piastra madre non occupato da alcun chip, ma che fa capo a certi indirizzi di porta e può essere quindi controllato via software. Ecco quindi che l'autore, con veramente una manciata di componenti, lo utilizza per far partire o fermare due registratori a cassette e ne suggerisce anche un potenziale utilizzo.

Prima di tutto l'I/O dei due registratori deve essere messo in parallelo in modo che poi ne sia interessato solo quello che sarà avviato dal software. Poi si pone un registratore in registrazione e l'altro in lettura.

Ad esempio se si volesse copiare dei dati da una cassetta sorgente ad una destinazione si comanderebbe la lettura facendo partire il relativo registratore, poi lo si ferma e si accende quello in scrittura, scaricandovi i dati tramite SAVE.

Semplice e ingegnoso, anche se all'oggi non può che far sorridere per la macchinosità necessaria ad una operazione tanto semplice, quale essa è divenuta di tutti i giorni: la copia di file.

La rivista, come altre del tempo, ospita certa pubblicità "facilona" che promette una rapida acquisizione delle tecniche di programmazione. Erano anni quelli, per chi li può ricordare, nei quali il lavoro era merce rara, non dico rarissima, ma rara. E soprattutto per i giovani che speravano di trovare nella tecnologia una strada non ancora chiusa dall'altrui presenza, in altri campi ormai abbondante.

Inutile dire che simili "scuole" non sono mai state grandi iniziative didattiche e sono servite veramente a pochi. Ma ne annotiamo il fenomeno quale testimonianza di una

CORSO RAPIDO DI PROGRAMMAZIONE SU MICRO COMPUTER

IN BASIC

GRATIS IN PROVA LA PRIMA DISPENSA PRINCIPALE

12 dispense principali + 12 dispense complementari

Se lei vorrà

Lei potrà

IST

società in fermento sulla soglia della nuova rivoluzione culturale.

Qui sotto ho voluto riportare una parte di pagina che riporta un listato ottenuto, come si è detto, da una copia di un LLIST dato sulla macchina. Per quanto abbia forzato la risoluzone al massimo, la scarsa qualità è del tutto evidente. I listati sono addirittura un po' "storti", quasi fossero stati fotocopiati senza eccessiva cura e trasferiti poi in stampa.

[Sn]

```
[illegible] GO SUB 8000
 410 LET d$="PREMI UN TASTO": L
ET y=4. LET ys=1. LET xs=2. GO S
UB 8000. LET d$="STOP per termi
nare". LET xs=1. LET ys=2. LET y
=[illegible] GO SUB 8000
 420 GO SUB 7000
 430 IF INKEY$="STOP" THEN GO
TO [illegible]
 440 LET [illegible] LET d$="Hai
premuto "+z$. LET y=12: GO SUB 8
000 GO TO 420
 450 BORDER 1: PAPER 5. CLS : LE
T d$="=================". LET xs=
2. LET ys=1. LET y=1. GO SUB 800
0. LET y=2. GO SUB 8000. LET d$=
"TEST CASSETTA ". BRIGHT 1. L
ET y=3. GO SUB 8000. BRIGHT 0
 460 DIM a(100). DIM b(100). FOR
i=1 TO 100. LET a(i)=RND. NEXT
i. LET d$="Metti il nastro in RE
CORD". LET ys=2. LET xs=1. LET y
=6. GO SUB 8000. LET d$="e premi
un tasto". LET y=8. GO SUB 8000
 470 LET d$="SCRITTURA". LET xs
=2. LET y=20 FLASH 1: INK 2. GO
SUB 8000. FLASH 0: SAVE "test" D
ATA a(). LET d$="
": LET y=20: GO SUB 8000. LET y
=6. GO SUB 8000 LET y=9. GO SUB
8000
 480 LET xs=1 LET d$="Ora riavv
olgi il nastro,". LET y=6: INK 2
 GO SUB 8000. LET d$="mettilo i
n PLAY e premi un tasto". LET y=
8: GO SUB 8000: LET d$="LETTURA"
 LET xs=2. LET y=20. GO SUB 700
0. FLASH 1 GO SUB 8000. FLASH 0
 LOAD "test" DATA b()
 490 LET d$="VERIFICA": LET y=20
 FLASH 1. GO SUB 8000: FLASH 0
 FOR i=1 TO 100. IF a(i)<>b(i) T
HEN GO TO 2000
 495 NEXT i
 500 LET d$="Registratore OK". L
ET xs=1. LET y=10. INK 0. GO SUB
8000. PAUSE 200
 510 BORDER 5. PAPER 1. CLS : IN
K 4 LET d$="Test Printer": LET
ys=1. LET xs=2. LET y=10 GO SUB
8000. LPRINT . LPRINT "Test LPR
INT"
 520 GO SUB 7500. INVERSE 1: GO
SUB 7500: INVERSE 0. LPRINT . LP
RINT "Test LLIST". LLIST 9000. L
PRINT . LPRINT "Test COPY". CLS
 530 INK 1 FOR i=20 TO 80 STEP
20 CIRCLE 128,88,i. NEXT i. PRI
NT [illegible]
[illegible] PRINT TAB 4;"
[illegible]" PRINT TA
B 11;"ZX PRINTER
 540 COPY . PAPER 7. BORDER 7. I
NK 0 LET xs=1 LET ys=2. LET d$
="Il test e terminato". LET y=3
. GO SUB 8000. LET d$="PREMI.".
[illegible] GO SUB 8000 LET d$="[illegible]
 per uscire" LET y=11. GO SUB
8000: LET d$="[illegible] per ripetere"
 LET y=15 GO SUB 8000
 550 GO SUB 7000 LET r$=INKEY$
 IF r$<>"e" AND r$<>"[illegible]" THEN GO
TO 550
 560 IF r$="r" THEN GO TO 85
 570 CLS . LET ys=1: LET xs=1. L
ET y=20 LET d$="© 1983 M. Pigot
t Research Ltd". GO SUB 8000
 580 PAUSE 0 STOP
 1000 CLS FOR i=32 TO -32 STEP
-1. BEEP ,05,i. NEXT i. FLASH 1
 LET d$="Test memoria fallito".
LET xs=1. LET ys=2. LET y=11: GO
 SUB 8000. FLASH 0 STOP
 2000 FOR i=1 TO 3. BEEP ,5,5: BE
EP ,5,0: NEXT i. CLS LET d$="V
ERIFICA FALLITA". LET y=11. FLAS
H 1 GO SUB 8000. FLASH 0. STOP
 7000 IF INKEY$="" THEN GO TO 700
0
 7005 BEEP .2 32
 7010 RETURN
 7500 FOR i=32 TO 164. LPRINT CHR
$ i; NEXT i. RETURN
 8000 LET y=8*y LET x=(256-xs*(
LEN d$))/2
 8000 LET a=23296. POKE a,x. POKE
 a+1,y POKE a+2,xs. POKE a+3,ys
 POKE a-4,ys LET a=a+4. FOR i=
1 TO LEN d$ POKE a+i,CODE d$(i)
 NEXT i. POKE a+i,255: RANDOMIZE
 USR 32256. RETURN
 9000 REM SAVE COMMANDS
 9010 SAVE "SPECTRUM" LINE 10
 9020 SAVE "schermo"SCREEN$
 9030 SAVE "codice" CODE 32256,344
 9040 STOP
```

Laboratorio

Interventi hardware e software per il ripristino, la manutenzione e l'evoluzione dei sistemi di calcolo personale.

Il computer ibrido



Ecco come si presenta la costruzione artigianale del calcolatore ibrido (immagine dalla rivista EPE).

Introduzione

Tutti noi, appassionati delle tecnologie digitali e della computazione in particolare, sappiamo come un calcolatore sia basato su una convenzione che prevede il trattamento di una grandezza fisica (tensione o corrente o anche altro) come portatrice di una unità di informazione: il famoso bit.

Come questo sia tradotto in grandezza fisica dipende dal "media" con il quale si ha a che fare. Ad esempio il livello di tensione +5 Volt come bit a uno e 0 Volt come bit a zero. A parte le tolleranze insite nel fatto che alla fine si tratta sempre di circuiti non perfettissimi (+5,1V vanno bene comunque), rimane il fatto che questa è una convenzione basata su uno standard industriale. Ma se invece che andare a misurare questa grandezza discreta sul piedino di uscita di un chip digitale per decidere se questo sia un uno o uno zero, potessimo misurare diciamo dieci differenti livelli di tensione ed attribuire ad essi la corrispondente cifra dell'aritmetica decimale? Allora l'informazione portata da una singola linea non sarebbe più binaria, ma decimale e per rappresentare un byte ci basterebbero (e ne avanza) tre unità invece che otto; un bel risparmio!

Se poi i valori discriminabili fossero 16 ci troveremmo una unità esadecimale belle e pronta e per un byte basterebbero due linee, due

unità elementari, due singole celle di memoria da un "bit" (continuiamo a chiamarlo così) cadauna.

Sembra l'uovo di colombo: immaginate che semplificazione circuitale se il bus indirizzi potesse trasportare per ogni linea sedici unità invece che due? Otto linee basterebbero per indirizzare 4 giga di indirizzi invece che le 32 necessarie con la numerazione binaria!

Ovviamente non è che la gente non ci abbia pensato, lo ha fatto eccome, anzi si è spinta addirittura più in la progettando circuiti elettronici che avrebbero dovuto permettere la costruzione di un calcolatore analogico capace di operazioni matematiche anche molto complesse con pochissimi elementi. Sono essi gli amplificatori operazionali e si chiamano proprio "operazionali" perchè sono costruiti al fine di compiere operazioni matematiche, le quattro elementari ma anche la derivata e l'integrale.

La gestione di una quantità continua e non più discreta, ad esempio tutti i livelli di tensione entro un certo range, comporta la possibilità di descrivere dei fenomeni fisici come grandezze continue ed avere delle risposte altrettanto continue dal sistema di calcolo.

Facciamo un esempio semplice. La somma di due quantità sarebbe eseguita da un circuito addizionatore il quale riceve in input due tensioni e restituisce in output la somma delle due. Nell'ambito della fisica classica la soluzione di problemi è risolvibile con una formula, oppure con l'integrazione di un sistema di equazioni oppure tramite differenziazione o integrazione. Niente di più facile da rappresentare con circuiti elettronici costruiti ad-hoc che usano gli operazionali.

La rinuncia nel proseguire questa strada per l'assemblaggio di macchine di calcolo è stata determinata da due fattori: l'impossibilità di gestire la tolleranza dei circuiti e la difficoltà di programmazione quando ancora il linguaggi per calcolatori erano solo nella mente dei pionieri. In realtà esiste un'altro ostacolo non indifferente che è il seguente: finché ci si limita a trattare le quantità a "run-time", cioè si lavora sulle grandezze in fase di calcolo, non ci sarebbero insormontabii problemi. Quando però si volesse cominciare a memorizzare queste quantità le cose si fanno se non più difficili sicuramente più onerose in termini di circuiteria e quindi di costo. Cosa è infatti memorizzare un bit

La costruzione di un simulatore di volo non avrebbe bisogno di calcoli digitali: le equazioni del moto deriverebbero direttamente dal probema fisico.



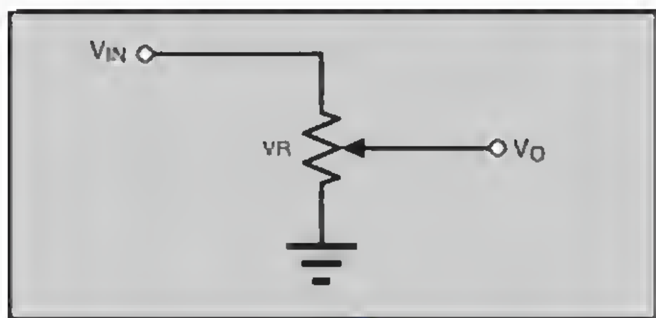


Figura 1

0/1 sfruttando la presenza/assenza di un valore, anche molto approssimato (si pensi ad esempio alle memorie dinamiche), e altro è progettare sistemi di memoria sia elettronici che magnetici che conservino non due ma 10 o anche sedici valori differenti.

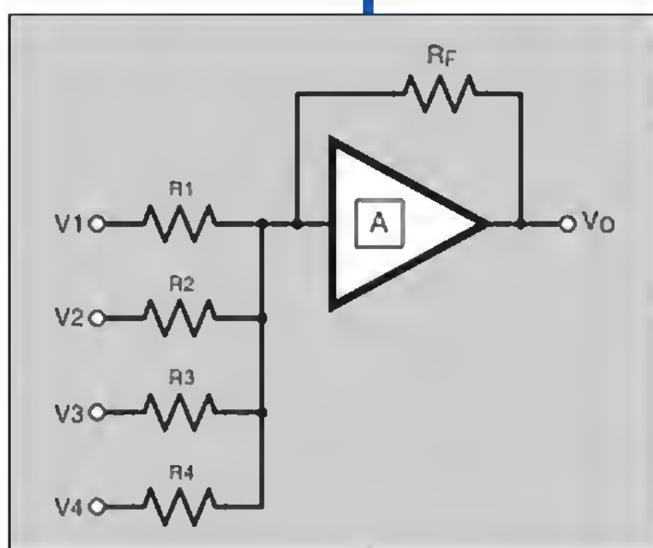
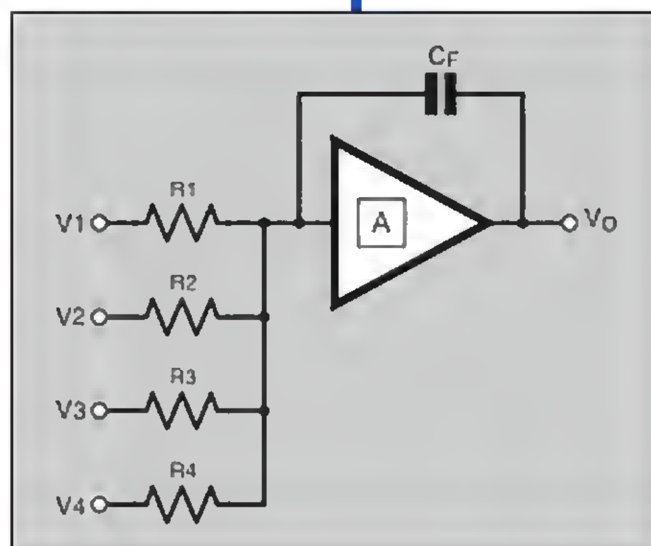


Figura 2

Questo non significa che non si possano costruire calcolatori analogici, la strada giusta è però un'altra: la costruzione di calcolatori ibridi, cioè macchine che sono in grado di gestire e calcolare quantità continue ma che possano essere programmate e restituire i valori ad un PC tradizionale.

Figura 3



Vediamo lo schema di questi circuiti di calcolo analogico.

Il più semplice da realizzare è il moltiplicatore per un coefficente (fig. 1). Esso è costituito da un semplice partitore di tensione che riceve in input la tensione Vin e restituisce una Vout che dipende dalla posizione del reostato R. Qualcuno obietterà che questo può moltiplicare solo per un fattore pari a 1 o inferiore, ed è vero; non è comunque un problema lavorare in un range e trasformare i risultato in un'altro, dal momento che ogni misura è di per sè una convenzione, vi pare? E' vero invece che l'ampiezza della tensione disponibile (da -15V a +15V) inevitabilmente limita la precisione del sistema e determina un valore massimo trattabile dal circuito.

Il circuito addizionatore fa uso di un amplificatore operazionale in configurazione "sommatore" (fig. 2). La tensione di uscita Vout è la somma delle tensioni di ingresso V1..V4. Elettronicamente il valore di uscita dipende anche dal rapporto fra la resistenza di feedback (reazione) e le resistenze a capo degli ingressi, secondo la formula: Vout = (Rf/R1*V1 + Rf/R2*V2 + Rf/R3*V3 + Rf/R4*V4).

Il circuito integratore usa un condensatore di reazione al posto della resistenza. Il valore dell'output è la somma degli integrali rispetto al tempo delle tensioni di ingresso V1..V4.

Ricordiamo che l'operazionale inverte il segno della tensione di ingresso ma un inverter è cosa semplice da costruire. Un'altra caratteristica degli schemi mostrati è

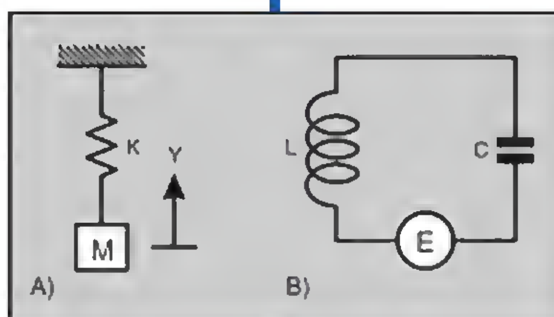
che sono presenti quattro ingressi anzicchè due, questo perchè lavorando sui valori delle componenti passive (le resistenze) è possibile predisporre due degli ingressi come moltiplicatori per 10 (o sommatori per 10) e con due si arriva alla moltiplicazione per 100. Ovviamente si potrebbero aggiungere ulteriori ingressi per praticare in catasta ulteriori operazioni sulle tensioni di ingresso, ma ad un certo punto per una costruzione pratica sarà necessario fermarsi ad un compromesso fra complessità circuitale e flessibilità dello stesso.

Costruire un calcolatore analogico è questione di disporre di un certo numero di questi circuiti, magari con la possibilità di predisporre il funzionamento di ciascuno secondo il bisogno, collegare gli input in cascata e leggere il risultato alla fine del "calcolo". Gli input sono tensioni e l'output pure, poi si tratterà di interpretarle in maniera coerente con il problema che si vuole comoputare. Predisposizione, collegamenti e impostazione dei parametri possono essere attività demandate a circuiti attuatori comandati da un PC via interfaccia e questo sarebbe l'ibridazione del calcolatore che da puro analogico diventa "hybrid".

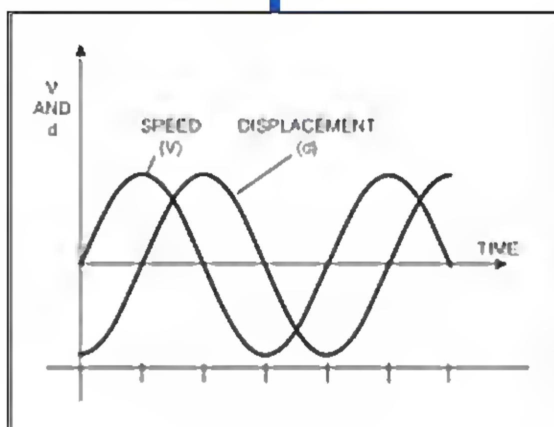
Quali i vantaggi di un calcolatore analogico, dal momento che gli svantaggi sono evidenti nella minore flessibilità dei problemi impostabili. La velocità di calcolo di un calcolatore analogico è praticamente quella della luce! Meglio: è quella della velocità di trasmissione dell'informazione elettromagnetica nei conduttori che lo formano, in ogni caso inarrivabile per qualsiasi macchina digitale costruibile ora e per sempre.

A questo punto è giusto che si faccia un esempio di uso. Supponiamo di voler risolvere un semplice problema fisico schematizzato dalla figura ... che consiste in un peso di massa M che oscilla sospeso ad una molla di costante K. Si tratta di ricavare il diagramma del moto della massa sospesa qualora si introduca una perturbazione di un allungamento della quantità D. Il diagramma sarà il grafico della posizione della massa M nel tempo.

Un problema del tutto analogo, ma se vogliamo più vicino al mondo dell'elettronica, è quello di un oscillatore composto da un generatore E, da un solenoide con impedenza L e un condensatore di capacità C.



Ecco i sistemi fisici che ci proponiamo di risolvere, cioè vogliamo ottenere le equazioni del moto nel tempo.



Un pendolo si sè, oscilla... bella scoperta. Velocità e distanza percorsa dalla massa sospesa alla molla sono sfasate di 90 gradi.

Per semplificare il sistema si suppone che non vi siano forze di attrito, che vedremo dopo come si possono introdurre. In queste condizioni il sistema effettua delle oscillazioni sinusoidali la cui ampiezza e periodo dipendono dalle grandezze in gioco.

*La massa M sospesa alla molla è soggetta alla legge di gravità e si trova inizialmente in una posizione che convenzionalmente chiamiamo zero. La massa viene tirata verso il basso di una quantità D e in queste condizioni la molla di costante elastica K tende a "tirarla" verso l'alto con una forza pari a K*D.*

Dal momento che le leggi del moto stabiliscono che:

*Forza = massa * accellerazione*

abbiamo:

*-K * D = M * A*

La forza è negativa, nel senso che richiama la massa verso la posizione di equilibrio perturbata dal nostro spostamento forzoso della massa verso il basso. L'incognita che ci manca è l'accellerazione A che possiamo ricavare molto semplicemente:

*A = -K * D / M*

La velocità della massa nel suo moto oscillatorio si ricava integrando l'accellerazione rispetto al tempo (infatti l'accellerazione è la derivata prima della velocità).

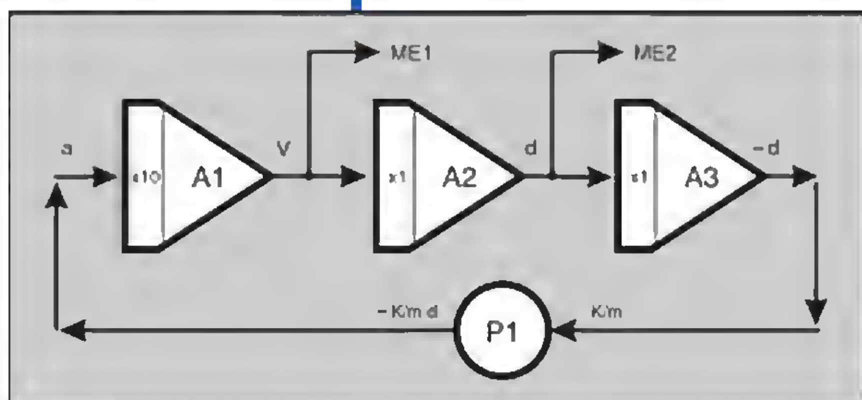
*V = Int(A) = Int(-K * D / M) dt*

La velocità V è la derivata dello spostamento rispetto al tempo, quindi lo spostamento si ottiene integrando la Velocità V. Il che è come dire che l'accellerazione è la derivata seconda dello spostamento rispetto al tempo.

Spero di non avervi fatto troppo soffrire con questo piccolo esercizio di fisica, quello a cui volevo arrivare è il fatto che disponendo del circuito che integra, si può ricostruire il grafico dello spostamento nel tempo della massa sospesa.

*La traduzione del problema in un circuito analogico è schematizzata nella figura a fianco; partiamo da un valore D della distanza impostando una certa tensione come ingresso di un inverter che ci restituisce -D (lo spostamento iniziale è negativo). Questa tensione la moltiplichiamo per un fattore costante K/M con un moltiplicatore ottenendo in uscita la grandezza -D*K/M che sappiamo essere l'accellerazione. Integriamo l'accellerazione con il circuito integratore OP(3) ottenendo la velocità V, misurata dal voltmetro S(1). Attraverso una ulteriore integrazione otteniamo la distanza D che diventa la nuova condizione di partenza.*

Per effettuare la simulazione (perchè tale si tratta) con un tradi-



Come vanno collegati i componenti analogici per calcolare le equazioni del moto. Me1 e Me2 sono due rivelatori (voltmetri), ma potrebbero essere due plotter che disegnano su carta, oppure semplicemente collegati all'input della parte digitale (un PC) del progetto.

zionale calcolatore, si procederebbe con un loop per avere la lista dei valori di velocità e distanza da "plottare" sul grafico. La velocità di calcolo digitale sarebbe legata alle prestazioni del calcolatore, per il circuito analogico la velocità è semplicemente quella della trasmissione dell'informazione da un punto all'altro del circuito.

Nella costruzione di un calcolatore ibrido le tecniche digitali entrerebbero nella comunicazione fra codice binario e gli attuatori circuitali, oltre che nella predisposizione delle grandezze in gioco e delle condizioni iniziali. Inoltre i voltmetri che misurano gli output sarebbero interfacciati con un convertitore e il PC sarebbe in grado di leggere i valori e agire di conseguenza. Il PC assume il compito di controllo e programmazione ma il calcolo "vero" sarebbe fatto dal circuito analogico predisposto.

So cosa state pensando: -"Mo' ti è andata bene perchè il problema era banale...". E' vero, ovviamente comporre la configurazione del circuito analogico in relazione al problema fisico da risolvere comporta un certo sforzo e capacità "programmatorie" diverse rispetto a quelle cui siamo abituati scrivendo delle righe di codice. Inoltre si presenterebbe presto il problema della limitatezza dei circuiti di calcolo disponibili. Non affermo quindi che programmare un circuito analogico sia più facile o possa risolvere qualsiasi tipo di problema, tutt'altro! Ci sono però problemi che un calcolatore tradizionare potrebbe non esser in grado di affrontare, mentre per il calcolatore analogico sarebbe questione di pigiare un bottone per dare il via alla simulazione e attendere che il circuito raggiunga l'equilibrio.

Conclusione

Abbiamo presentato in questo articolo i principi generali e una ipotesi prototipale di realizzazione di un calcolatore analogico. Chi volesse approfondire l'argomento potrà trovare due articoli relativi alla costruzione di un kit sui numeri di Novembre e Dicembre della rivista Everyday Practical Electronics del 2002 che mette a disposizione anche il Kit per la realizzazione del prototipo composto da dieci amplificatori operazionali.

[Mg]

Bibliografia

"Introduction to Analogue Computers", Technical Education & Management Inc., Foulsham. ISBN 05720027895

"Introduction to Electronics Analogue Computers", C. A. A. Wass, Kenneth Charles Gardner, Pergamon Press, ISBN 0080110711.

Biblioteca

I testi di riferimento di ieri e di oggi.

La pratica dell'Apple



Scheda

Titolo: *La pratica dell'Apple*

Sottotitolo: La grafica, il sistema Apple, il BASIC Applesoft

Autore: Nicole Bréaud Pouliquen

Editore: Gruppo editoriale Jackson

Anno: 1981

ISBN: 86-7056-145-3

Lingua: lItaliano

Pagine: 130

I primi sistemi home erano degli emeriti sconosciuti per quasi tutti coloro che si avvicinavano alla micro informatica. L'incontro avveniva normalmente nelle aule delle facoltà scientifiche o come pratica hobbistica derivata dalla passione elettronica. Nel 1981 pochissimi potevano dire di avere visto un Apple][e ancora meno coloro che potevano vantarsi di averci messo le mani sopra. Serviva una letteratura molto basilare ma che soddisfacesse nel contempo quella curiosità tecnica cui i primi possessori delle macchine d'epoca non potevano sottrarsi.

Il volume "La pratica dell'Apple" si rivolge a quella ampia schiera di utilizzatori che cercano di capire ogni dettaglio del sistema che hanno difronte, vogliono conoscerne i segreti, carpire i trucchi e con essi realizzare programmi migliori o semplicemente stupire amici e conoscenti con le meraviglie della programmazione.

In un agile volumetto di nemmeno 130 pagine, sono condensate una marea di utili informazioni che spaziano dall'utilizzo del BASIC Applesoft, in dotazione ai sistemi della mela, alle mappe di memoria e di I/O, fino alla mappatura e ai timing dei segnali sul bus di espansione.

Il volume è organizzato in tre capitoli e sei appendici. Nella prima parte viene descritta l'organizzazione tecnica della macchina e i comandi del monitor. Nel secondo capitolo viene esaminato in dettaglio il software BASIC Applesoft con la desamina delle istruzioni ed alcuni esempi di utilizzo. Una parte è dedicata alla pratica del PEEK e POKE, ponte di collegamento fra BASIC e linguaggio macchina. Per ultimo si affronta in maniera esau-

sita le capacità grafiche del sistema: un campo dove la curiosità era massima e le possibilità molto limitate, ma esaltanti al tempo stesso.

Nelle appendici trovano posto le tabelle utili durante lo sviluppo delle applicazioni: le parole riservate del Basic, il set dei caratteri, la mappatura della memoria, etc…

Conclusione

Un utile testo che permette in poche pagine di acquisire le nozioni base e più che base del sistema Apple][. Terminata la sua funzione educational, oggi potrebbe risultare utilissimo per qualcuno che intenda avvicinarsi ad uno dei sistemi simbolo del retro computing.

Nicole Breaud-Pouliquen è un ingegnere consulente in informatica "personale". In questo ruolo integra programmazione da parecchi anni. Pouliquen utilizza l'Apple II da quando questo calcolatore è stato commercializzato in Francia e s'interessa, più specificatamente, dell'utilizzo del personal computer nell'insegnamento.

L'autore ringrazia per il prezioso lavoro svolto nella stesura dell'edizione italiana la signora Francesca di Fiore e l'ing. Roberto Pancaldi

Traduzione a cura di Piero Dell'Orco.

Stampato in Italia da:
S.p.A. Alberto Matarelli - Milano - Stabilimento Grafico

II

SOMMARIO

III

IV

INTRODUZIONE

Cominciate ad utilizzare il vostro Apple per gradi, e vi renderete conto di quali sono le sue risorse senza spiacevoli inconvenienti.

E' ben noto il linguaggio Basic nelle sue tradizionali funzioni e sono sovente utilizzate istruzioni del tipo "paracadute" che sembrano spesso misteriose. Dedicando un pò di tempo a questo problema sarà possibile comprendere la ragion d'essere di queste istruzioni.

Questa andatura, che il libro pone come obbiettivo, permetterà di riscoprire le possibilità dell'Apple II. Si potranno scrivere programmi con cognizione di causa; sarà possibile spiegare meglio ad amici neofiti il significato di Apple ed avvicinarli al computer con spirito scientifico ed analitico.

Il libro comprende tre capitoli, Il Sistema Apple II, il linguaggio Applesoft, i disegni e la grafica. Gli argomenti sono trattati con il preciso intento di far prendere coscienza delle reali possibilità dello strumento (numeri reali, stringhe di caratteri, loro codifica interna, loro occupazione in memoria etc.)

Si consiglia di effettuare una prima lettura superficiale, al fine di selezionare gli argomenti che interessano in modo specifico e di elaborare un metodo per poi assimilarli nel modo più corretto.

V

[Sm]

Retro Linguaggi



ABAP (parte 4)

La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.

Istruzione CASE

Riprendiamo dal punto dove ci eravamo lasciati la volta scorsa per esaminare un'altra importante istruzione del linguaggio: CASE.

Questa permette la costruzione di branch strutturati con la seguente costruzione sintattica:

```
CASE <campo>.
      WHEN <valore]>.
      istruzioni
      WHEN <valore2>.
      istruzioni
      WHEN <valore3>.
      istruzioni
      WHEN OTHERS.
      istruzioni
ENDCASE.
```

Box 1.
Esempio uso dell'istruzione CASE

```
* ---branching with CASE

DATA STRINGA TYPE STRING.
MOVE 'A' to STRINGA.
DATA: TEXT1 VALUE 'X',
      TEXT2 VALUE 'Y',
      TEXT3 VALUE 'Z'.


CASE STRINGA.
      WHEN TEXT1 OR TEXT2.
            WRITE: / 'String is',
                   TEXT1, 'OR', TEXT2.
      WHEN TEXT3.
            WRITE: / 'String is', TEXT3.
      WHEN OTHERS.
      WRITE: / 'String is not', TEXT1,
              TEXT2, TEXT3.
ENDCASE.

Move 'Y' to STRINGA.
CASE STRING.
      WHEN TEXT1 OR TEXT2.
            WRITE: / 'String is',
                   TEXT1, 'OR', TEXT2.
      WHEN TEXT3.
            WRITE: / 'String is', TEXT3.
      WHEN OTHERS.
            WRITE: / 'String is not',

                   TEXT1, TEXT2, TEXT3.
ENDCASE.
```

Con l'utilizzo della case, soltanto una delle sequenze di istruzioni verrà eseguita. L'istruzione when others è opzionale, mentre solo un valore può essere testato nella when; questo può essere sia un numero che il nome di una variabile.

L'esempio riportato nel Box 1 chiarisce l'utilizzo di questa importante istruzione.

Alcune operazioni di stringa

La capacità di trattare le stringhe è una caratteristica importante per tutti i linguaggi e in particolare per quelli orientati al business.

Il sorgente riportato nel Box 2 mostra alcune operazioni classiche sulle stringhe. Ne andiamo ad esaminare qualche particolarità.

TRASLATE è l'istruzione che converte in maiuscolo o minuscolo una stringa. La semplice operazione di conversione globale della stringa in uno o nell'altro caso è banale e non la spieghiamo qui. Ci sofferminalo invece in un'altra capacità del linguaggio, quella di definire dei "pattern" di conversione da usare per l'operazione di tra-

sformazione.

Ci riferimao alla determinazione di una regola di conversione, qui definita nell'istruzione:

```
MOVE 'AxByFz' TO REGOLA_DI_
CONVERSIONE.
```

Da usarsi poi come template nell'istruzione di conversione vera e propria.

Un'altra possibilità di elaborazione delle stringhe sono le operazioni raggruppate nell'insieme "pattern matching" (riportate nella seconda parte del listato).

L'operazione più classica è la SEARCH che permette, come dice il nome, di cercare un pattern all'interno di una stringa.

Facciamo notare che nel listato abbiamo usato una funzione di impaginazione dellla WRITE che permette di incollonare dei dati indicando con "UNDER" sotto quale intestazione essi devono essere stampati.

In effetti le possibilità di impaginare un listato è una delle forze dell'ABAP, che porta all'estremo i tentativi già presenti nei linguaggi prientati al business come nel COBOL o nell'RPG.

L'incolonnamento funziona nel seguente modo: si scrivono le intestazioni di colonna che nell'esempio abbiamo valorizzato nelle tre parole 'Searched', 'SY-SUBRC' e 'SY-FDPOS', poi in ogni istruzione WRITE successiva si può aggiungere la direttiva "UNDER <nome-colonna>" che indica in quale colonna (individuata dal nome dell'intestazione) si intende avere l'output.

Nell'elaborazione di stringhe è utile il campo di sistema SY-FDPOS (field-position) che viene valorizzato al valore corrispondente ad

```
* --- conversione in maiuscolo/minuscolo

DATA testo TYPE string.
MOVE 'Conversione Maiuscolo e/o Miniscolo'
      TO testo.
WRITE: /, 'testo originale = ', testo.
TRANSLATE testo TO UPPER CASE.
WRITE: /, 'Tutto maiuscolo = ', testo.
TRANSLATE testo TO LOWER CASE.
WRITE: /, 'Tutto minuscolo = ', testo.

* --- conversione di caratteri nelle stringhe

DATA TESTO_DA_CONVERTIRE TYPE STRING.
DATA REGOLA_DI_CONVERSIONE TYPE STRING.
MOVE 'ABCDEF' TO TESTO_DA_CONVERTIRE.
MOVE 'AxByFz' TO REGOLA_DI_CONVERSIONE.

TRANSLATE TESTO_DA_CONVERTIRE
          USING REGOLA_DI_CONVERSIONE.
WRITE: /, TESTO_DA_CONVERTIRE.


* --- calcolo lunghezza di una stringa
DATA LUNGHEZZA TYPE I.

LUNGHEZZA = STRLEN( TESTO_DA_CONVERTIRE ).
WRITE: /, LUNGHEZZA.


* --- pattern matching

DATA STRING(30) VALUE 'This is a little sentence.'.

WRITE: / 'Searched', 'SY-SUBRC', 'SY-FDPOS'.
ULINE /1(26).

SEARCH STRING FOR 'X'.
WRITE: / 'X', SY-SUBRC UNDER 'SY-SUBRC',
              SY-FDPOS UNDER 'SY-FDPOS'.

SEARCH STRING FOR 'itt '.
WRITE: / 'itt   ', SY-SUBRC UNDER 'SY-SUBRC',
                   SY-FDPOS UNDER 'SY-FDPOS'.

SEARCH STRING FOR '.e .'.
WRITE: / '.e .', SY-SUBRC UNDER 'SY-SUBRC',
       SY-FDPOS UNDER 'SY-FDPOS'.

SEARCH STRING FOR '*e'.
WRITE: / '*e ', SY-SUBRC UNDER 'SY-SUBRC',
       SY-FDPOS UNDER 'SY-FDPOS'.

SEARCH STRING FOR 's*'.
WRITE: / 's* ', SY-SUBRC UNDER 'SY-SUBRC',
       SY-FDPOS UNDER 'SY-FDPOS'.
```

Box 2.
String operations e pattern matching

```
Searched SY-SUBRC SY-FDPOS

X          4     0
itt        0     11
.e .       0     15
*e         0     10
s*         0     17
```

Figura 1. Output incolonnato con il risultato della ricerca di pattern.

una posizione per tutte le funzioni che hanno questo tipo di compito.

Nel caso dell'istruzione SEARCH, l'indice SY-FDPOS conterrà l'indice della prima occorrenza della stringa cercata.

L'output del codice di esempio è riportato nella figura 1 in questa pagina.

Nell'esempio sono mostrate alcune delle caratteristiche per la modulazione della ricerca. '*e' significa "una parola che termina con 'e'; 's*' è la ricerca per una parola che inizia con 's'; '.e.' è la sintassi per la ricerca di una sottostringa all'interno di un periodo.

Le caratteristiche del linguaggio ABAP sono molto avanzate nell'istruzione SEARCH così come nelle altre istruzioni del linguaggio. Ad esempio si veda l'interessante caratteristica seguente:

```
SEARCH 'Sognando la Califor-
nia.' FOR 'Clfrn' ABBREVIATED.
```

L'istruzione permette di cercare una stringa anche se non se ne conosce l'esatta composizione. Nell'esempio le lettere 'Clfrn' sono contenute nella parola 'California' e pertanto la ricerca avrà risultato positivo.

L'istruzione CONDENSE interviene su una stringa eliminando spazi o caratteri di separazione non desiderati.

Il codice determina l'output visibile nella figura seguente.

```
 one two three four
Length:          19

one two three four
Length:          18

onetwothreefour
Length:          15
```

Box 3. Funzione CONDENSE

```
* --- CONDENSATE STRING

DATA: STRING(25) VALUE ' one two three four ',
LEN TYPE I.

LEN = STRLEN( STRING ).
WRITE: STRING, '!'.
WRITE: / 'Length: ', LEN.

CONDENSE STRING.
LEN = STRLEN( STRING ).
WRITE: STRING, '!'.
WRITE: / 'Length: ', LEN.

CONDENSE STRING NO-GAPS.
LEN = STRLEN( STRING ).
WRITE: STRING, '!'.
WRITE: / 'Length: ', LEN.
```

La variabile STRING è dichiarata di 25 caratteri e riempita con delle parole separate da uno spazio. La stringa del valore ha anche uno spazio all'inizio e uno alla fine.

La funzione che calcola la lunghezza della stringa restituisce il valore 19 che è la lunghezza della stringa senza gli spazi in coda.

L'operazione "CONDENSE" elabora la stringa togliendo gli spazi iniziali (trailing spaces). Il successivo calcolo della lunghezza mostra correttamente il valore di 18.

Infine la specifica "NO-GAPS" toglie anche gli spazi intermedi nella stringa.

Nell'esempio appena concluso è stata dichiarata una variabile con nome STRING anche se si tratta di una parola riservata. A differenza di altri linguaggi ABAP distingue il contesto. Sono legali, anche se possono generare confusione, dichiarazioni del tipo:

```
DATA I TYPE I.
DATA D TYPE D.
DATA DATA TYPE D.
```

Infine due istruzioni fondamentali: la divisione di una stringa in "token" e la composizione di più stringhe in una sola.

Si veda nel Box 4 un esempio di uso della funzione SPLIT per dividere una stringa nelle sue componenti.

In questo esempio la variabile STRING2 viene inizializzata con una stringa contenente dei caratteri (tre asterischi) che saranno usati come delimitatori. Lo statement SPLIT divide la stringa considerando il delimitatore nelle variabili P1, P2, P3 e P4.

L'operazione inversa è la concatenazione di stringhe che si effettua con l'istruzione CONCATENATE.

```
CONCATENATE P1 P2 P3 P4
            INTO STRING2.
```

Si noterà che la funzione di concatenazione inserisce uno spazio fra le stringhe concatenate.

Una operazione meno comune è quella svolta dall'istruzione OVERLAY che permette una sovrapposiione fra due stringhe. Vediamo un esempio riportato nel box 4.

```
* --- SPLIT STRING

DATA: STRING2(60),
P1(20) VALUE '++++++++++++++++++++',
P2(20) VALUE '++++++++++++++++++++',
P3(20) VALUE '++++++++++++++++++++',
P4(20) VALUE '++++++++++++++++++++',
DEL(3) VALUE '***'.

STRING2 = ' Part 1 *** Part 2 *** Part 3 *** Part 4
*** Part 5'.
WRITE STRING2.

SPLIT STRING2 AT DEL INTO P1 P2 P3 P4.

WRITE / P1.
WRITE / P2.
WRITE / P3.
WRITE / P4.

* --- OVERLAY

DATA: T(10) VALUE 'a c e g i ',
      STRING3 LIKE T,
      OVER(10) VALUE 'ABCDEFGHIJ',
      STR(2) VALUE 'ai'.

STRING3 = T.
WRITE STRING3.
WRITE / OVER.

OVERLAY STRING3 WITH OVER.
WRITE / STRING3.

OVERLAY STRING3 WITH OVER ONLY STR.
WRITE / STRING3.
```

Box 4.
Divisione di stringhe

STRING3 viene inizializzata con il valore 'a c e g i '; OVER è un'altra stringa che contiene la sequenza di caratteri 'ABCDEFGHIJ'.

La prima istruzione overlay modifica il contenuto di STRING3 inserendo nelle posizioni occupate inizialmente dagli spazi il corrispondente carattere che si trova nella stringa OVER.

Se l'istruzione viene ripetuta con la specifica ONLY:

Solo le lettere 'a' e 'I' vengono interessate alla sostituzione.

[Mx]

BBS

Posta e comunicazioni

A colloquio con i lettori

E-mail

da insonnia67.

Intervengo sull'articolo "i migliori PC di tutti i tempi" che ho letto dal numero 9. Ho capito che voi non c'entrate nulla e che anzi avete cercato di correggere le più evidenti anomalie, mi chiedo solo come hanno fatto i redattori originali a dimenticare macchine come il Commodore 64 e lo Spectrum! I casi sono due: o questi signori sono in mala fede, non capisco però quale possa essere lo scopo, oppure sono persone impreparate che hanno sparato qualcosa così tanto per fare ma che non conoscono il dettaglio della storia della microinformatica degli ultimi trent'anni. Mi può anche stare bene un Apple II al primo posto, ma il Commodore64 per diffusione, capacità elaborativa, biblioteca software, etc... deve stare al secondo posto!

Che ne pensate?

PS: approfitto per complimentarmi della vostra iniziativa. Perché non mettete sul sito una sezione con le vostre foto, così vi conosceremmo meglio?

Risponde Sm

Carissimo/a insonnia67, vedi che anche tu ami l'anonimato e il nickname che Internet non solo rende possibile ma anche incoraggia? Se facessimo una sezione del sito con foto, curriculum e scheda personale, chi assicurerebbe che il tutto sia reale? In fondo l'anonimia, per chi ritiene sia giusto praticarla, non danneggia nessuno. Però una pagina per redattore con una scheda un po' più dettagliata ce l'hanno richiesta anche altri e non è detto che non si faccia.

Venendo al merito era ovvio, come del resto aveva anticipato l'articolo, che la pubblicazione della classifica avrebbe mosso proteste e precisazioni. Del resto qualsiasi lista in ordine di merito è destinata ad essere parziale. C'è da dire che i criteri adottati dai redattori potevano forse salvaguardare dagli errori più evidenti come quello che tu citi (il C64 è comunque menzionato fra i secondi 25 della classifica).

Credo a ragione che se facessimo noi un sondaggio e una relativa classifica si otterrebbe ancora una versione parzialissima del merito dei singoli progetti. La maggior parte delle persone voterebbe per il "suo" sistema e solo pochi, anche per ragioni di conoscenza, indicherebbero una lista obiettiva di sistemi.

Continua a seguirci, ciao.

Rassegna

L'andamento del mercato dei retro calcolatori è ovvio sia nel mirino della nostra attenzione. E' il sito eBay che fa in pratica le quotazioni e che tutti tengono monitorato per avere una indicazione obiettiva del valore degli apparati che intendono vendere o comprare.

Trattandosi di una rassegna di proposte di vendita e di rilanci sugli acquisti, non si tratta di una quotazione reale, infatti è soggetta alle leggi dell'economia "percettiva", quella che regola i nostri acquisti di impulso e voluttuari. Infatti una cosa è comprare beni necessari (come il pane) o di largo consumo (come la benzina) e un'altra è quella di decidere l'acquisto di un retro calcolatore. In quest'ultimo caso infatti il valore dell'oggetto è solo parzialmente determinato dall'effettivo valore e invece per la maggior parte dettato dalla nostra cupidigia nel possederlo. A chi serve "veramente" un Commodore 64? O uno Spectrum o una workstation SGS? Evidentemente a nessuno. Chi compra lo fa per soddisfare un suo piacere: possedere quella macchina che è stata il suo compagno di infanzia, ri-giocare un vecchio videogioco che in passato ci ha tenuti incollati al Joystick per intere settimane, etc...

Da un paio di anni in qua si sta assistendo ad un fenomeno inpensabile: le quotazioni degli apparecchi sono in crescita. Se due anni fa si poteva comprare uno Spectrum Plus+2 per venti euro, ora a meno di quaranta non si riesce ad aggiudicarselo.

Ancora più interlocutoria è la quotazione del Commodore64. Un sistema completo di lettore floppy (senza la macchina vale pochino) si prende sui 50 Euro mentre in passato 25-30 Euro erano la norma.

Aldilà dei meccanismi economici che non vengono affatto smentiti dalla situazione attuale (più domanda = prezzo più alto), sarebbe interessante analizzare l'aspetto sociale. Come mai si vendono ancora e bene dei Commodore 64, quando tutti i retro computeristi d'Italia già ne possiedono più di un esemplare? Il numero di appassionati è in crescita? Stanno andando in pensione coloro che la macchina l'hanno posseduta in passato e vogliono rinnovare i loro ricordi?

Fateci sapere come la pensate.

Errata/Corrige

Bruno Grampa, titolare del sito "gli amici di HAL" ci chiede di precisare la titolarietà del sito zxspectrum.hal.varese.it a carico di Stefano Guida.
Nel fascicolo n. 15 di JN a pagina 36, l'introduzione dell'articolo sulla fanzine ZXNOTIZIE attribuisce la paternità dell'iniziativa al gruppo con sede hal.varese.it. Bruno ci invita a precisare che hal.varese.it ospita il sito web dell'iniziativa ma che non è direttamente collegato ad essa, essendo la fanzine e il sito opera del solo Stefano.
Prontamente contattato Stefano ci ha autorizzato alla pubblicazione del suo nome quale autore del sito e della rivista ZXNotizie.
Ci scusiamo con Stefano per l'involontaria svista e auguriamo allo stesso di riprendere quanto prima la pubblicazione di questa interessantissima fanzine.